

ENDÜSTRİYEL OKULLAR İÇİN

**NC MAKİNE
PROGRAMCILIĞI
VE
PROGRAM TASARIMI**

Endüstriyel Okullar İçin **NC Makine Programcılığı ve Program Tasarımı**

Satış fiyatı: 149.000 Lira
KDV: 1.490
KDV'li SATIŞ FİYATI: 150.490

TOPTAN SATIŞ

İstanbul Devlet Kitapları Müdürlüğü, Adana, Ankara, Burdur, Elazığ,
Erzurum, İzmir, Samsun, Sivas, Trabzon, Van ve Zonguldak
Bölge Şeflikleri.

PERAKENDE SATIŞ

Millî Eğitim Yaynevleri ve Bakanlık yayınları satıcısı kitapçılar.

2704

65

30 01 97
F.B

ENDÜSTRİYEL OKULLAR İÇİN

NC Makine Programcılığı ve Program Tasarımı

Chao-Hwa Chang
Michel A. Melkanoff



Evren Ofset A.Ş. Web Ofset Tesisleri ANKARA - 1994

Millî Eğitim Bakanlığı Yayınları : 2704
Yardımcı ve Kaynak Kitaplar Dizisi : 65

ISBN 975 - 11 - 0878 - 0

Hükümetimiz ile Dünya Bankası arasında imzalanan Endüstriyel Okullar Projesi çerçevesinde hazırlanan "NC Makine Programcılığı ve Program Tasarımı" Millî Eğitim Bakanlığı, Talim ve Terbiye Kurulu Başkanlığının 04/05/1994 gün ve 3680 sayılı kararı ile kaynak kitap olarak uygun bulunmuş ve 20.000 adet bastırılmıştır.

Çeviri - Dizgi - Mizanpaj : Ünlversal Dil Hizmetleri ve Yayıncılık A.Ş.
Çevirmen : Dr. Mahmut GÜLESİN
Öğretim Görev. Ersan ASLAN
Editör : Doç. Dr. Muammer NALBANT
Baskı Hazırlık - Baskı - Cilt : Evren Ofset Basım Sanayii ve Ticaret A.Ş.

© Yayın hakkı: Prentice-Hall, Inc.
Türkçe yayın hakkı Millî Eğitim Bakanlığına aittir. 1994



İSTİKLÂL MARŞI

Korkma, sönmez bu şafaklarda yüzen al sancak,
Sönmeden yurdumun üstünde tüten en son ocak.
O benim milletimin yıldızıdır, parlayacak,
O benimdir, o benim milletimindir ancak.

Çatma, kurban olayım çehreni ey nazlı hilâl!
Kahraman ırkıma bir gül! Ne bu şiddet, bu celâl!
Sana olmaz dökülen kanlarımız sonra helâl...
Hakkıdır, Hakk'a tapan, milletimin istiklâl!

Ben ezelden beridir hür yaşadım, hür yaşarım.
Hangi çılgın bana zincir vuracakmış? Şaşarım!
Kükremiş sel gibiyim, bendimi çiğner, aşarım.
Yurtarım dağları, enginlere sığmam, taşarım.

Garbın âfâkını sarmışsa çelik zırhlı duvar,
Benim iman dolu göğsüm gibi serhaddim var.
Ulusun, korkma! Nasıl böyle bir imanı boğar,
"Medeniyet!" dediğin tek dişi kalmış canavar?

Arkadaş! Yurduma alçakları uğratma, sakın.
Siper et gövdeni, dursun bu hayâsızca akın.
Doğacaktır sana va'dettiği günler Hakk'ın...
Kim bilir, belki yarın, belki yarından da yakın.

Bastığın yerleri "toprak" diyerek geçme, tanı :
Düşün altındaki binlerce kefensiz yatanı.
Sen şehit oğlusun, incitme, yazıktır, atanı :
Verme, dünyaları alsan da, bu cennet vatanı.

Kim bu cennet vatanın uğruna olmaz ki fedâ?
Şühedâ fişkıracak toprağı sıksan, şühedâ!
Canı, canânı, bütün varımı alsın da Huda,
Etmesin tek vatanımdan beni dünyada cüdâ.

Ruhumun senden, İlahî, şudur ancak emeli :
Değmesin mabedimin göğsüne nâmahrem eli.
"Bu ezanlar-ki şahâdetleri dinin temeli-
Ebedî yurdumun üstünde benim inlemeli.

O zaman vecd ile bin secde eder-varsa-taşım,
Her yerihandan, İlahî, boşanıp kanlı yaşım,
Fişkırr ruh-ı mücerred gibi yerden nâ'sım;
O zaman yükselerek arşa değer belki başım.

Dalgalar sen de şafaklar gibi ey şanlı hilâl!
Olsun artık dökülen kanlarımın hepsi helâl.
Ebediyen sana yok, ırkıma yok izmihlâl:
Hakkıdır, hür yaşamış, bayrağımın hürriyet;
Hakkıdır, Hakk'a tapan, milletimin istiklâl!

Mehmet Âkif ERSOY



ATATÜRK'ÜN GENÇLİĞE HİTABESİ

Ey Türk gençliği! Birinci vazifen, Türk istiklâlini, Türk cumhuriyetini, ilelebet, muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin, en kıymetli hazinendir. İstikbalde dahi, seni, bu hazineden, mahrum etmek isteyecek, dahili ve harici, bedhahların olacaktır. Bir gün, istiklâl ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için, içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin! Bu imkân ve şerait, çok nâmûsait bir mahiyette tezahür edebilir. İstiklâl ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın, bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elîm ve daha vahim olmak üzere, memleketin dahilinde, iktidara sahip olanlar gaflet ve dalâlet ve hattâ hıyanet içinde bulunabilirler. Hattâ bu iktidar sahipleri şahsî menfaatlerini, müstevlilerin siyasi emelleriyle tevhid edebilirler. Millet, fakr u zaruret içinde harap ve bîtap düşmüş olabilir.

Ey Türk istikbalinin evlâdı! İşte, bu ahval ve şerait içinde dahi, vazifen; Türk istiklâl ve cumhuriyetini kurtarmaktır! Muhtaç olduğun kudret, damarlarındaki asil kanda, mevcuttur!

K. Atatürk

Bilgi çağına girerken bütün ülkelerin üzerinde önemle durdukları ve giderek daha fazla kaynak ayırdıkları sektör eğitimidir. Bilim ve teknolojideki gelişmelere paralel olarak eğitimde kaliteyi yükseltmek, gençlerimize ileri sanayi toplumunun gerektirdiği bilgi, beceri ve davranışları kazandırmak Millî Eğitimimizin temel amaçlarından biridir.

Ülkemizde; ekonomik, sosyal ve kültürel alanlarda olduğu gibi, sanayi alanında da önemli gelişmeler olmaktadır. Nitelikli insangücü ihtiyacının giderek arttığı ülkemizde meslekî ve teknik eğitim büyük önem kazanmaktadır.

Bu alandaki ihtiyacı karşılayabilmek için; çağdaş bilim ve teknolojik metodları bilen, yorumlayan, kullanan, geliştiren ve alanındaki yeniliklere uyum sağlayan, üretken teknik insangücünün yetiştirilmesi gerekmektedir. Bu konuda, teknik öğretim kurumlarımıza büyük iş düşmektedir.

Bu kurumlarımızdaki öğrencilerin iyi yetişmeleri için devletimiz her türlü desteği sağlamakta ve Hükümetimiz ile Dünya Bankası arasında imzalanan İkraz Anlaşmasıyla yürütülen Endüstriyel Okullar Projesiyle bu okullarımız, çağdaş eğitim imkanlarına kavuşturulmaktadır. Bu okullarımızda çeşitli meslek alanlarında ihtiyaç duyulan 42 adet yabancı teknik ders kitabının tercüme haklarının satın alınması, basım ve dağıtımlarının yapılarak öğrenci ve öğretmenlerimizin istifadesine sunulması, bu proje kapsamında yürütülen faaliyetlerden biridir.

Eğitim ve kültür düzeyleri yüksek, gelişen teknolojiye uyum sağlayabilen toplumlar, geleceğin dünyasının şekillenmesinde önemli rol oynayacaklardır.

Bu ve benzeri çalışmaların ülkemiz için yararlı olmasını diliyorum.

Nevzat AYZ
Millî Eğitim Bakanı

SUNUŞ

Varlıklarını sürdürmek isteyen toplumlar, kalkınmanın gerektirdiği sayıda nitelikli insangücünü yetiştirmek için eğitime değer vermek ve ona bilimsel ve teknolojik bir nitelik kazandırmak mecburiyetindedirler.

Eğitim, Cumhuriyetin kuruluşundan beri ülkemizde yenileşme aracı olarak görülmüştür. Bugün Eğitim sistemimiz, bilim çağına girilen dünyamızda, toplumumuzun büyüyen ve çeşitlenen ihtiyaçlarına cevap vermede bir takım problemlerle karşı karşıyadır.

Eğitimle ilgili problemlerin çözümünde, yeni yöntemler, teknikler ve araçlar geliştirmek için araştırmalar yapmak, ayrıca daha önce yapılmış araştırmalar sonucu geliştirilen bilgi ve teknolojiyi ülkemize getirmek zorundayız.

Eğitime ayrılacak finansman kaynaklarının sınırlı olması, ülkemizi, genel bütçe dışındaki imkanlardan faydalanmaya zorlamaktadır. Devletimiz bu imkanları araştırmış, mesleki ve teknik öğretim kurumlarımızın bilim ve teknolojiye meydana gelen gelişmelere paralel olarak modernleştirilmesi için Uluslararası İmar ve Kalkınma Bankası (Dünya Bankası - IBRD) ile yapılan ikraz Anlaşmasıyla Endüstriyel Okullar Projesi uygulamaya konulmuştur.

Bu projenin amaçları; Endüstriyel Okulların yeni teknoloji ürünü makina ve teçhizatla donatılarak yenilenmesi, çeşitli meslek alanlarında müfredat programlarının geliştirilmesi, burslar ve yurt dışından danışman temin edilmesi yoluyla öğretmenlerimizin eğitilmesi ve çeşitli meslek alanlarında ders kitaplarının tercüme ve yayın haklarının satın alınarak Eğitim Sistemimize kazandırılmasıdır.

Proje ile belirlenen hedeflere büyük ölçüde ulaşılmıştır. Projenin amaçlarından biri olan çeşitli meslek alanlarında (Hidrolik - Pnömatik, Soğutma ve İklimlendirme, CNC, Döküm, Elektronik, Bilgisayar, PLC ve Metal İşleri) teknik ders kitapları, uzmanlardan kurulu komisyonlarca seçilmiş ve tercüme edilerek yayımlanmıştır.

Büyük kaynak ve emek harcayarak Eğitim Sistemimize kazandırdığımız kitapların öğretmen ve öğrencilerimize faydalı olmasını dilerim.

Salih ÇELİK
Projeler Koordinasyon
Kuruğu Başkanı

önsöz

İkinci Dünya Savaşından bu yana, mühendislik eğitimindeki eğilim, temel ilkelerin ve tasarımın önemini vurgulamaktadır. Üretime, Amerikan Üniversitelerinde mühendislik bilimi ile ilgili olmaktan çok bilgiye dayalı bir teknoloji olarak sayıldığı için daha az önem verilmiştir. Üretim aynı şekilde sanayi tarafından da ürünün performansını geliştirme açısından ve hatta maliyetleri düşürme açısından da ikinci sırada kabul edilmiştir. Böyle bir düşünce aşağıdaki bakış açısı tarafından yansıtılmaktadır: "Üretim maliyetlerindeki tasarrufların ve masrafları engelleyici etkenlerin en belirginini üretim yöntemlerindeki veya sistemlerindeki değişikliklerden çok ürünün tasarımındaki değişikliklerdir."* Bu, bir tasarımın ötekinden bariz olarak daha mantıklı olduğu zaman doğrudur. Bununla beraber, belli bir ürünün iki tasarımı benzer veya karşılaştırılabilirse, her bir ürünün performansını, kalitesini ve maliyetini üretim işlemleri ve üreticinin yönetimi belirler. Bu, dünya pazarındaki payları sürekli olarak artmakta olan ve son 15 sene içerisindeki üretim yöntemleri ve yönetimindeki dikkate değer gelişmelerin sonucu, Japon endüstrilerinin başarıları ile kendini fazlası ile göstermiştir. Amerikan endüstrisi üretimin hayati önemini, ki artık hiç göz ardı edilemeyecektir, çok acı tecrübelerle öğrenmiş durumdadır. Bu eğilim değişikliğinin önemli bir göstergesi, Amerikan (ve yabancı) otomobil endüstrilerinin "bütün dikkatlerini" şimdi "üretime çevirmiş" olmalarıdır.+

Üretimi ve bizim ona bakış açımızı belirgin olarak değiştiren önemli bir etken, bilgisayarların gelişmesi ve yeni nesil üretim ekipmanları, özellikle bilgisayarlı sayısal denetimli (CNC) tezgahlardır.

CNC tezgahlar, robotlar, bilgisayarlar, CAD/CAM sistemleri, esnek üretim sistemleri (FMS) ve bilgisayar-bütünlüklü üretim sistemlerini (CIMS) kapsayan programlanabilir cihazların ve sistemlerin her çeşidinin giderek artan kullanımı ile üretim tamamen devrim geçirmiştir. Bilgisayar destekli üretim (CAM) bir parça üretilmeden önce her işlem kademesinin net olarak tanımlanmasını ve her detayın planlanmasını gerektirir. Bu üretim işleminin, yıllar önce işlem şemasında veya genel işlem planında yapıldığı gibi bir çok detayın operatöre veya teknisyene bırakıldığı gibi gelişigüzel tanımlanamayacağı anlamına gelmektedir. Diğer kelimelerle ifade edilirse, üretim artık bundan böyle çağdaş CAD/CAM ortamı içerisinde uygulamaya bağlı bir şekilde basitçe ele alınamaz. Üretim yönetimi, işlem planlanması, parça sınıflandırılması, otomatik parça ve malzeme yönetimi, otomatik kontrol etme, takım tezgahının durumunun otomatik izlenmesi, bilgisayarların üretim ekipmanları ile bütünlüğü, üretim veri tabanları, üretim uygulama yazılım-

larının tasarımı ve hatta üretim işleminde bile daha fazla araştırma gereklidir. Yeni üretim mühendislerine temel mühendislik bilgilerine ek olarak üretim işlemi hakkında kapsamlı bilgi, üretim yönetimi ve yeni bilgisayarların ve yeni nesil üretim ekipmanlarının uygulamaları öğretilmelidir. Böyle bir araştırma ve eğitim programı bütün mühendislik okullarının aktif katılımını gerektirir.

Yeni durumu göğüsleyebilmek ve endüstrinin acil ihtiyaçlarına cevap verebilmek için Los Angeles'taki Kaliforniya Üniversitesi (UCLA) 1980'de Üretim Mühendisliği programını kurdu. Çağdaş bilgisayar destekli tasarım "NC tezgahların bilgisayarla doğrudan evliliğinin büyümesi" olarak sayılabileceğinden bizim programımızda sunulan kurslarda NC her zaman en üst sıralarda idi. NC kurslarını vermeye hazırlandığımızda, İngilizce yayınlanmış NC kitapları olmadığını görünce şaşırдық. Bundan başka bulabildiğimiz kitapların çoğu NC teknolojisinin genel tanıtımı ve elle NC programlama ile ilgili idi. Bilgisayar destekli NC programlama, CAD/CAM sistemleri ile NC programlama, sonişlemci tasarımı, NC programının doğruluğunun kontrolü ve NC tezgah-denetim ünitelerinin tasarımı gibi bir çok önemli konular ya işlenmemişti ya da sadece çok az değinilmişti. Bu kitaplar üniversite seviyesinde ders kitapları olarak muhtemelen hizmet veremeyeceklerdi. Geçmiş yıllar boyunca UCLA'da gerçekleştirilen iki NC kursunda öğretim için kullanılan bu ders notları sonradan hazırlandı. Üretim mühendisliği programları ve NC kursları eğitimi veren bir çok üniversitede çok acil iyi NC ders kitaplarına ihtiyaç vardır.

Bu kitap bu acil ihtiyaca cevap vermek üzere yazılmıştır. Bizim NC Tezgahların Programlanması ve Kontrol Yazılım Tasarımı ve Sayısal Kontrollü Tezgah Laboratuvarı isimli iki NC kurstaki tecrübelerimiz üzerine kurulmuştur. Bu kitap NC tezgahların programlanmasını değişik yaklaşımlardan son işleme için tasarlanmasına ve uyarlanmasına kadar olan aşamaları sistemli bir şekilde tartışır. APT dilini kullanarak programlamaya, CAD/CAM sistemleri ile NC programlamaya ve sonişlemci tasarımına önem verilmiştir. Modern CNC tezgahların elle programlanması tartışılmasına rağmen bu kısım sadece kitabın geri kalanını anlamaya yetecek kadar uzatılmıştır.

Bu kitap ya lisans düzeyinde üst bölümlerde NC ders kitabı olarak ya da NC ve APT programcıları, üretim mühendisleri ve NC yazılım tasarımcıları için başvuru kitabı olarak düşünülmüştür. Lisans veya lisans üstü düzeyde bir CAM kursunda destekleyici ders kitabı olarak kullanılabilir. Normalde bu kitapta yer alan materyalin çoğu derslerden ve laboratuvar çalışmalarından oluşan iki kurs halinde sunulmuştur.

Kitap 15 bölüme sahip olup 4 ana kısma ayrılmıştır;

- I. Giriş ve Tezgah-Okuyabilir Kodlarla Elle NC programlama
- II. APT Dilinde Bilgisayar Destekli NC Programlama

- III. NC Programlarının CAD/CAM sistemi ile türetilmesi
- IV. Sonişlemcilerin Tasarımı ve Uyarlanması

İlk iki kısım çok az programlama alt yapısına sahip okuyucuların anlayabileceği şekilde yazılmıştır. Okuyucuların Kısım III'te yer alan tartışmaları izleyebilmesi için CAD sistemleri ile bazı tecrübelerinin olması tercih edilir. Kısım IV'ü anlamak için FORTRAN programlanmasının bilinmesi gereklidir.

Kısım I iki bölümden oluşur. Bölüm 1, NC üzerindeki alt yapıyı vermekte ve NC teknolojisinin çağdaş üretim mühendisliğine etkisini anlatmaktadır. Bölüm 2, NC tezgah kodlarını, bir NC programının bloklarının formatını ve modern CNC tezgahların elle programlama yöntemlerini anlatır.

Kısım II'de APT ile bilgisayar destekli NC programlamayı tartışacağız. APT dilini ve onun programlama ilkelerini tarif etmek için bu dilin ilk tasarlandığı sistem olmasından dolayı IBM sistemi seçilmiştir. Programlama detaylarını açıklamak için çok sayıda örnek verilmiştir. Bu kısım altı bölüme ayrılmıştır. Bölüm 3, APT dilinin elemanlarını tanıtır ve sözlük kelimelerini, sayıları ve skalarları, noktalama ve sınırlayıcıları ve deyim etiketlerini kapsar.

Bölüm 4, kesici hareketlerini tanımlamanın temel elemanları olan geometrik öğelerin tanımları ile ilgilidir. Değişik geometrik öğeleri tanımlamak için gereken deyimlerin formatlarını ve kullanımları detaylı olarak tanımlanmıştır.

Bölüm 5 kesici hareketlerinin tanımlarını tarif eder ve notadan-noktaya, çevreleme hareketi ve bir desen şeklinde tanımlanmış nokta gruplarının hareket sıralarını kapsar.

Bölüm 6, işleme özelliklerini ve tezgah işlemlerinin tanımlanmasında kullanılan deyimleri açıklar.

Bölüm 7, kesici yollarının ve tezgah işlemlerinin programlanmasının basitleştirilmesi için kullanılan deyimleri ve yöntemleri tanıtır. Döngü programlama, alt programlarla programlama ve kesici yollarının ve geometrik öğelerin dönüşümü tartışılmıştır. Ayrıca boşaltma yordamını tanımlayan deyim kullanımı tarif edilmiştir.

Bölüm 8, APT programının genel yapısını, hatalarının giderilmesini ve APT programının IBM APT-AC Sayısal Denetim İşlemcisi tarafından işlenmesini açıklar.

NC programlarının CAD/CAM sistemleri kullanılarak türetilmesi Kısım II-I'te açıklanmıştır. CAD modeline dayanarak NC program türetilmesinin altında yatan ilkeler ve CADAM sisteminde kesici yollarının tanımlanmasında kullanılan yöntem ve kurallar karşılıklı olarak Bölüm 9 ve Bölüm 10'da tarif edilmiştir. Bilgisayar destekli NC programlamanın gerçekleştirilmesi için, sonişlemciler (otomatik NC programlama sistemlerinden, APT veya CAD/CAM sistemlerinden gelen çıktılarını NC tezgahı tarafından okunabilir kodlara dönüştürmek için kullanı-

lan yazılım) gereklidir. Tasarımı ve uyarlanması detaylı olarak Kısım IV'te tartışılmıştır. Yine IBM, sistemi derinlemesine anlatmak için seçilmiş olmasına rağmen kullanıcının kendi sisteminde sonişlemci tasarlayabilmesi amacı ile bir sonişlemcinin tasarımının genel ilkelerine ve uyarlanmasına önem verilmiştir. Sonişlemci ile bir tanışma ve bilgisayar destekli NC programlama sisteminden elde edilmiş bir çıktı (örneğin CLDATA) Bölüm 11'de verilmiştir. Bölüm 12'de sonişlemcinin içindeki haberleşme araçlarını ve yöntemlerini tartışacağız. Tezgahtan bağımsız genel işlem yordamları Bölüm 13'te ve tezgaha bağımlı işleme yordamları Bölüm 14'te açıklanmıştır. Bölüm 15 bir sonişlemcinin oluşturulması, tasarımı, testi ve uyarlanması işlemlerini tarif etmektedir. Bir sonişlemci örneği detaylı olarak tarif edilmiştir.

NC sonişlemcilerindeki eğilimler üzerine bir tartışma bu kısmın son sözünde yer almaktadır.

NC programlamanın ve sonişlemci tasarımının bu kadar kapsamlı bir tartışması, verilen NC tezgah kontrollerine ve bilgisayar programlama sistemlerine ilişki kurulmadan anlatılması çok zor olacaktır. Bu yüzden biz, en yaygın kullanılan gerçek sistemleri seçtik. Bununla beraber, programlamanın esaslarına ve yazılım tasarımına önem verilmiştir ki, bunlar diğer sistemlere de uygulanabilir.

Bir ders kitabı olarak yapılan iş kaçınılmaz olarak diğerlerinin işleri üzerine yaratılmıştır. Biz IBM Corporation'na kendi yayım hakları olan malzemeleri kullanma izni verdikleri için teşekkür ediyoruz. CADAM INC'a sağladıkları fotoğraflar ve çizimler için ve malzemelerini öneren diğer firmalara da özel teşekkürlerimizi sunmak istiyoruz. Son olarak yapıcı önerileri için Prof. Yoram KOREN'e içten teşekkürlerimizi sunmak istiyoruz.

C.-H. Chang ve M.A. Melkanoff

İçindekiler

KISIM 1 / TEZGAH-OKUYABİLİR KODLARLA NC PROGRAMLAMA 1

BÖLÜM 1 /Nümerik Kontrol Teknolojisine Giriş, 3

- 1.1 NC Teknolojisinin Kısa Tarihçesi, 3
- 1.2 NC Teknolojisinin Bugünkü Durumu, 11
- 1.3 NC Teknolojisinin Tasarım ve Üretim Üzerine Etkisi ,13 Sorular, 15

BÖLÜM 2 / Tezgah İşlemlerinin Elle NC Programlanmasına Giriş, 17

- 2.1 NC'li Tezgahlarının Koordinat Sistemleri, 19
- 2.2 Tezgah Kodu, 20
- 2.3 Bir NC Programının Blok Formatları, 21
- 2.4 Tornalama ve Frezeleme İşlemleri için NC Programlamada Kullanılan Temel Kodlar, 27
- 2.5 Koordinat Sisteminin Tanımlanması, 32
- 2.6 Kesici Hareketlerinin Tanımlanması, 35
- 2.7 İşleme Çevrimlerini ve Tekrarlamalı İşleme Çevrimlerini, Tanımlayan Kodlar, 48
- 2.8 Yardımcı Fonksiyonlar Tanımlayan Kodlar, 67
- 2.9 NC Programlama Örnekleri, 69 Problemler, 81

KISIM 1 İÇİN KAYNAKLAR, 93

KISIM 2 / APT DİLİNDE BİLGİSAYAR DESTEKLİ NC PROGRAMLAMA 95**BÖLÜM 3 / APT Dilinin Temel Elemanları, 98**

- 3.1 APT Kelimeleri, 99
- 3.2 Semboller, 102
- 3.3 Sayılar ve Skalalar, 102
- 3.4 Noktalama İşaretleri ve Sınırlayıcılar, 103
- 3.5 Deyim Etiketleri, 111
- 3.6 Deyim Boyutu Sınırlaması, 112
- 3.7 APT Deyim Formatı İçin Gösterim, 112

BÖLÜM 4 / Geometrik Öğelerin Tanımlanması, 114

- 4.1 Bir Noktayı Tanımlayan Deyimler, 120
- 4.2 Bir Doğruyu Tanımlayan Deyimler, 129
- 4.3 Bir Daireyi Tanımlayan Deyimler, 141
- 4.4 Bir Vektörü Tanımlayan Deyimler, 155
- 4.5 Düzlemsel Eğrileri (Elips, Hiperbol, G Konik (GCONIC) ve L Konik (LCONIC)) Tanımlayan Deyimler, 162
- 4.6 Bir Düzlemi Tanımlayan Deyimler, 165
- 4.7 İkinci Dereceden Yüzeyleri, Silindireleri ve Döner Yüzeyleri Tanımlayan Deyimler, 173
- 4.8 Sınırlandırılmış Yüzeyleri Tanımlayan Deyimler, 188
- 4.9 Çok Konilli Bir Yüzeyi Tanımlayan Deyimler, 194
- 4.10 Noktaların ZSURF Deyimiyle Tanımlanması, 199
- 4.11 APT Geometrik Tanım Deyimleri Üzerine Bazı Yorumlar, 201
- 4.12 Matematiksel Hesaplama Fonksiyonları, 206
- 4.13 Geometrik Tanımlama Deyimlerinin Uygulamaları, 210
Problemler, 216

BÖLÜM 5 / Kesici Hareketinin Tanımlanması 225

- 5.1 Noktadan-Noktaya Hareketin Tanımlanması, 226
- 5.2 Bir Noktalar Dizisi İçin Noktadan-Noktaya Hareketin Tanımlanması : PATTERN Deyimi, 228
- 5.3 Bir Engelden Sakınmak veya Kaçmak İçin Kullanılan İlave Kesici Hareketini Tanımlayan Değiştiriciler, 248
- 5.4 Çevresel İşleme Hareketinin Tanımlanması, 254

- 5.5 Bir Çevresel İşleme Hareketini Başlatma ve Bitirme, 264
- 5.6 Bir Çevresel Hareketi İçin Parça Yüzeyinin Tanımlanması, 272
- 5.7 Kaba ve İnce Çevresel İşleme Hareketlerinin Tanımlanması, 273
- 5.8 Çevresel İşleme Hareketinin Programlanması, 274
- 5.9 Çevresel İşleme Hareketi Deyimlerinin İşlenmesi, 278
Problemler, 283

BÖLÜM 6 / İşleme Özellikleri ve Çeşitli Deyimler, 292

- 6.1 Toleransları Tanımlayan Deyimler, 292
- 6.2 Ölçü Birimini Tanımlayan Deyimler, 294
- 6.3 Kesici Aleti Tanımlayan Deyimler, 295
- 6.4 İlerleme Hızını Tanımlayan Deyimler, 303
- 6.5 İş Mili Dönüş Hareketini Tanımlayan Deyimler, 304
- 6.6 Kesici Takımı Değiştirmek İçin Kullanılan Deyimler, 304
- 6.7 Soğutucuyu Tanımlayan Deyimler, 306
- 6.8 İşleme Esnasında Beklemeyi Tanımlayan Deyim, 307
- 6.9 Dairesel Hareketi Tanımlayan Deyim, 307
- 6.10 Tezgaah İşleminin Sonunu Belirten Deyim, 308
- 6.11 Tezgaah Koordinat Sistemini Tanımlayan Deyim, 308
- 6.12 İşlemci Denetim Kelimeleri, 310
- 6.13 Yorum Metnini Belirleyen Ek Deyimler, 312
Problemler, 314

BÖLÜM 7 / Geometrik Öğelerin ve Kesici Hareketinin Daha Karmaşık Deyimlerle Tanımlanması, 316

- 7.1 İndisli Değişkenler ve Kontrol Dallarına Deyimleri, 316
- 7.2 Döngü Programlama 322
- 7.3 Alt Programlar İle Programlama (MACRO) 332
- 7.4 Döngü ve Alt Programlardaki Yaygın Programlama Hataları 334
- 7.5 Geometrik Öğeleri ve Kesici Yolunu Tanımlamak için Matematik Hesaplama Deyimlerinin Döngüler ve/veya Alt Programlarla Kullanılması, 337
- 7.6 Dönüşüm Matrislerini Tanımlayan Deyimler, 339
- 7.7 Dönüşüm Matrisi İle Geometrik Öğelerin ve Kesici Yollarının Tanımlanması, 349
- 7.8 Başlatma ve Çevresel İşleme Hareketlerini Tanımlayan Diğer Deyimler, 364
- 7.9 POCKET (Cep) Deyimi 370
Problemler 377

BÖLÜM 8 / Bir APT Programının Yapısı Ve İşlenmesi 383

- 8.1 Tornalama İşlemlerinin Programlanması İçin Koordinat Sistemi, 384
- 8.2 Son İşlemcinin APT Programlanmasına Etkisi, 384
- 8.3 Bir APT Programının Genel Yapısı, 386
- 8.4 APT Program Örnekleri, 387
- 8.5 IBM APT-AC Nümerik Kontrol İşlemcisine Giriş, 398
- 8.6 Bir APT Programının Çalıştırılması İçin İşlem Sırası, 401
- 8.7 Bir APT Programından Hata Ayıklama, 404
- 8.8 İşlem Sırası Listeleme Dosyası ve CLDATA 409
Problemler, 412

Part II İçin Başvuru Kaynakları 374**KISIM 3 / NC PROGRAMLARININ CAD/CAM SİSTEMİ İLE
TÜRETİLMESİ, 375****BÖLÜM 9 / CAD/CAM Sistemlerinde NC Programlama İlkeleri, 423**

- 9.1 CAD/CAM Sistemlerine Giriş 423
- 9.2 Bir CAD Modeline Bağlı NC Programı Oluşumunun İlkeleri, 428
- 9.3 CAD/CAM Sistemi İçinde Kesici Hareketinin Oluşturulması, 430
- 9.4 NC Parça Programlarının Doğrudan CAD Modelinden Otomatik Oluşumuna
Bağlı Dikkat Edilmesi Gereken Hususlar, 436
- 9.5 Sorular, 438

BÖLÜM 10 / CADAM Sistemi ile NC Programlarının Türetimi, 392

- 10.1 CADAM Sisteminde Bir NC Programının Türetimi İçin İşlem Sırası Ve
Kontrol Akışı, 442
- 10.2 Tezgah İşlemlerinin Ve Kesici Yolunun Ekranda Tanımlanması İçin
Yöntemler ve İşlem Sırası, 444
- 10.3 Bir NC Programının Bilgisayardan NC Denetleyicisine Gönderilmesi, 458

Kısım III İçin Başvuru Kaynakları, 459**KISIM 4 / IBM APT-AC NUMERİK KONTROL İŞLEMCİSİNE GÖRE SON
İŞLEMCİLERİN TASARIMI VE UYARLANMASI, 462****BÖLÜM 11 / Sonişlemciler ve CLDATA'ya Giriş 463**

- 11.1 Sonişlemcinin Gerekliği, 463
- 11.2 CLFILE Dosyası, 464
- 11.3 Bir Sonişlemcinin Genel Yapısı, 469
- 11.4 Sonişlemcinin Fonksiyonları, 474
Problemler, 476

**BÖLÜM 12 / DAPP-Tabanlı Sonişlemcide ki Temel Değişkenler Ve İletişim
Kanalları, 477**

- 12.1 DAPP'ta Kullanılan Temel Dizi Ve Değişkenler, 478
- 12.2 DAPP-Tabanlı Sonişlemcideki Değişkenlere İlk Değerlerin Atanması: DAPP
Soru Listesi, 490
- 12.3 "BLOCK DATA" Alt Programını Hazırlamak İçin Kullanılan İşlem Sırası, 500
Problemler, 508

BÖLÜM 13 / DAPP-Tabanlı Sonişlemcinin Tezgah- Bağımsız İşleme Yordamları, 510

- 13.1 İlk Değer Atama Bölümü, 510
- 13.2 Düzenleme Bölümü, 512
- 13.3 Okuyucu Bölümü, 514
- 13.4 Genel İşleme Bölümü, 516
- 13.5 Çıktı Bölümü, 544
Problemler, 545

BÖLÜM 14 / Tezgah Bağımlı İşleme Yordamı: Takım Tezgahı Modülü (MTM) 546

- 14.1 Bir Takım Tezgahı Modülünün (MTM) Genel Yapısı, 547
- 14.2 MTM'nin İlk Değer Verme Kısım, 551
- 14.3 MTM'nin İşleme Kısım, 553
- 14.4 MTM'nin Hazırlama ve Kontrol-Yönlendirme Bölümleri, 575

14.5 DAPP-Tabanlı Sonişlemcide İşlem Akışı, 575
Problemler, 577

BÖLÜM 15 / DAPP-Tabanlı Son Bir İşlemcinin Oluşturulması: Bir Örnek 580

15.1 DAPP-Tabanlı Sonişlemcinin Oluşturması İçin Kullanılan İşlem Sırası, 580
15.2 CADILLAC NC-100 torna/FANUC 6T Denetim Sistem İçin DAPP Tabanlı Bir
Son İşlemcinin Tasarımı ve Uygulaması, 582
Problemler, 612

Kısım 4 İçin Son Söz 619

Kısım 4 İçin Kaynaklar 623

EKLER 625

A APT Dilinde Kullanılan Kelimeler, 626
B APT de Tanımlanan Seçilmiş Geometrik Öğelerin Kuralına Uygun Biçimleri, 629
C Çizelgelenmiş Bir Silindir İçin Çıktının Formatı ve Kapsamı, 630
D DAPP-Tabanlı Bir Son İşlemcisinin Oluşumu ve OS/MVS Ortamında Bir APT
Programının İşlenmesi için İş Denetim Dili (100) Yazılımları, 633

İNDEKS 635

Kısım 1

**Tezgah Okuyabilir Kodlarla
NC Programlama**

Bölüm 1

Nümerik Kontrol Teknolojisine Giriş

Nümerik Kontrol (NC) tezgahların ve işlemlerin sayılardan, harfler ve diğer sembollerden oluşan kodlarla otomatik olarak çalıştırılmasına izin veren bir tekniktir. NC teknolojisi başlangıçta metal kesme tezgahlarının otomatik kontrolü için geliştirilmiş olmasına rağmen, uygulamaları çok değişik tezgahlara ve işlemlere uzatılmıştır. NC teknolojisinin en önemli katkılarından biri, tezgahların otomasyon şeklini değiştirmesidir. NC'e bağlı tezgah otomasyonu değişik üretim durumlarına kolayca uygulanabilir ve ayarlanabilir. Bilgisayar teknolojisinin uygulamaları ile beraber NC, çağdaş bilgisayar destekli üretime (CAM) kapıları açar ve gelecekteki insansız otomasyon ve üretim sistemlerini gerçekleştirecek temelleri sağlar.

1.1 NC TEKNOLOJİSİNİN KISA TARİHÇESİ

1950'lerden önce üretim endüstrisinde iki farklı üretim yöntemi vardı: Küçük veya orta ölçekli üretim ki bu elle üretilen, düşük üretim hızına sahip çok çeşitli parça ve ürünlerdir. Diğeri ise yüksek hacimli, otomatize edilmiş üretim, ki bunda belli parça tiplerini yüksek miktarlarda tutarlı bir kalitede yüksek hızlarda üretebilmektir. Bunun için özel tasarlanmış veya ayarlanmış takım tezgahları kullanıldı. Sonraki yaklaşımın uyarlanması her zaman makinalara, aperlara ve yardımcı donanımlara oldukça büyük sermaye yatırılması gündeme gelmiştir. Sadece yapılacak parçaların sayısı yüksek ilk sermaye yatırımı karşılayacak kadar çok sayıda olan durumlarda kendini haklı çıkarmıştır. Bu iki yaklaşımın üretim kapasiteleri arasında çok keskin bir ayrım vardı. Örnek olarak, elle kumanda edilen vida makinalarında bir günde sadece birkaç yüz vida yapılabilirken, otomatik olanlarda ise günde binlerce üretilirdi. Otomatik makina veya sistemlerin belli türde parçalar için tasarlanmasından dolayı otomatik üretim sistemlerinin veya makinalarının başka çeşit parçaları yapacak şekilde ayarlanabilmesi çok zor, hatta imkansızdı.

İkinci Dünya Savaşından bu yana değişen talep, teknolojik gelişme ve uluslararası rekabet geçmişte olduğundan çok daha hızlı adımlarla yeni ürün tasarımlarının ortaya çıkmasına yol açtı. Bir ürün kalitesinde, özelliklerinde ve performansında yükselme olmadan ve başka bir deyişle tasarımında değişiklikler yapılmadan uzun süre varolamaz. Çoğu parça için parça tasarımında mümkün olduğu kadar az değişiklik gerektiren eski otomatik üretim yöntemleri haklı çıkmaktadır. Eski türdeki otomatik takım tezgahı veya üretim sistemi, üzerlerinde değişiklik yapılması çok zor olan mekanik, elektromekanik, pnömatik veya hidrolik donanım sistemleri ile kontrol edilen işlemler tarafından karakterize edilirdi. Örnek olarak, klasik otomatik takım tezgahı makina üzerinde tasarlanan hareket sıralarını ve işlemleri gerçekleştirmek için kamları, tamburları, mekanik sınırlayıcıları, sınırlama anahtarlarını ve şablonları kullanırdı. Parça tasarımındaki herhangi bir değişiklik kamların, tamburların ve şablonların değişmesini ve mekanik sınırlayıcıları, sınırlama anahtarlarının yeniden ayarlanmasını gerektirirdi. Bazı durumlarda otomatik takım tezgahı yeni bir parçayı üretmek için hiç bir şekilde kullanılamazdı. Bundan dolayı yeni ilkelere dayalı ve parça tasarımındaki ve üretim koşullarındaki değişikliklere kolayca uyulanabilecek yeni bir tür kontrol sistemi gereksinimi vardı.

Yeni denetim sistemi aynı şekilde insan tarafından kesintiye uğratılmadan kesici takım konumlarının daha yüksek hassasiyetle denetlenmesini gerektirmekte idi. İkinci Dünya Savaşından sonra yüksek performanslı uçaklar ve otomobiller gibi daha fazla sayıda ürünler, hidrolik kopya frezeleme tezgahlarında yapılması çok zor olan ve çok zaman gerektiren karmaşık profilli ve şekilli parçalara sahipti. Kesici takımın çevresel işleme hareketlerini yüksek hassasiyetle yapmasını sağlayacak şekilde sinyalleri yüksek hızlarda işleyebilecek yeni bir tür kontrol sistemine ihtiyaç vardı. İkinci Dünya Savaşının sonuna doğru öncekilerden yüzlerce kez hızlı çalışan ilk sayısal elektronik bilgisayarın gelişimi ile böyle bir yeni tür denetim sisteminin yapılması ihtimali elde edildi.

Yeni bir denetim sisteminin getirilmesini doğrudan yardım eden şey İkinci Dünya Savaşından sonra tasarlanan yeni yüksek performanslı bir savaş uçağına duyulan gereksinimdi. Bu uçakların üretimindeki en önemli problem kopya frezeleme tezgahlarının kapasitesini çok aşan hassas çevreleme işlemlerine gerek duyulması idi. Üstelik tasarımlar devamlı olarak değiştirilmekte ve geliştirilmekteydi. Amerikan Hava Kuvvetleri metallerin geliştirilmiş bir üretkenlikle hassas olarak işlenebilmelerini sağlayacak yeni bir teknolojinin geliştirilmesinin zorunluluğunun farkına vardı.

İkinci Dünya Savaşı sırasında Parsons Corporation bir freze tezgahının tablasını boylamasına ve enlemesine eşzamanlı olarak iki operatörün yardımı ile hareket ettirebilmek için hesaplanmış işleme koordinat tabloları (kesici konumu sayısal verileri gibi) kullandı.¹ Parsons Corporation'dan Jonh Parson çevrelenmiş

parçaların işlemlerindeki deneyimine dayanarak üç boyutlu kesici konumlarının çevrelenmiş parçalar için üretilebileceğini ve daha sonra takım tezgahının hareketlerini kontrol etmek için kullanılabileceğini önerdi. Onun bu önerisi ve bu zamana kadar başarılan teknoloji ile çok geniş bir ufka sahip olan William T. Webster ve Hava Malzeme Komutanlığındaki bir kaç mühendis yeni yüksek hassasiyetli çevreleme tezgahının teknolojisi için sayısal bir bilgisayar ile yüksek performanslı bir servo mekanizmanın bütünleştirilmesi gerektiği sonucuna vardılar.²

Bilgisayar kontrollu bir takım tezgahının fizibilitesini çalışmak için ilk kontrat Ekim 1949'da Parsons Corporation'a verildi, ki o da Massachusetts Institute of Technology (MIT)'in Servomekanizma Laboratuvarını alt kontratla kullanacaktı. MIT'in çalışması yüksek hassasiyetli çevreleme tezgahlarının gerektirdiği performans karşılayabilecek sistemin fizibilitesini onayladı. İlk yeni denetim sistemi tarafından kontrol edilen eş zamanlı üç eksen harekete sahip düşey freze tezgahı MIT tarafından 1952'de imal edildi. Analog-sayısal karışımı bir denetim ünitesine sahipti ve işleme programını (sayısal kodlu tezgah emirlerinin sıralanması) saklamak için ikili sistemde hazırlanmış delikli şerit kullanılıyordu ve nümerik kontrollü tezgah olarak adlandırıldı.

Bu tezgah öncekilere göre üç veya beş kat daha fazla hassasiyet ve tekrarlanabilirlikte ve daha yüksek hızlarla bir parçanın yapılabileceğini gösterdi.² Ek olarak yeni bir parçayı üretmek için ne şablon ne de makina elemanlarında bir değişiklik gerekli idi: sadece delikli şeritte saklı program yeniden hazırlanması gereken tek şeydi.

1952 ile 1955 boyunca Amerikan Hava Kuvvetleri tarafından verilen bir kontratla MIT yeni NC tezgah kontrol sisteminin denenmesi ve onun diğer takım tezgahlarına uygulanması ile ilgili ek araştırmalar gerçekleştirdi. Amerikan Hava Kuvvetleri endüstrideki kişilerin gerekli NC programını hazırlamalarının zor olacağını tahmin ettiğinden NC programlama tekniklerinin de geliştirilmesi de araştırmanın başka bir önemli konusu idi. Aynı şekilde bu yeni teknolojinin endüstriye aktarılabilmesi için MIT'in tanıtım yapmak zorunda olduğu da planlandı. Bununla beraber bu son amaç başarılı olmadı. Elektronik, sayısal denetim, yüksek hassasiyetli ölçme ve programlama gibi bir çok endüstriyel alandaki gelişmelerden dolayı hiç bir firma NC sistemi ısmarlamak veya üretmek için istekli olmadı. Bundan sonra Hava Kuvvetleri 1959'da, 60 milyon dolarlık parçalarının işlenmesi için 100 adet büyük NC freze tezgahı yapımının maliyetini kendisi finanse etmeye karar verdi. İşe yaramayan sistemlerin ortaya çıkması riskini azaltmak için denetleyici dört ayrı gruptaki tezgah yapımcısı-kontrol üreticilerine verildi.²

- Kearney ve Tracker Bendix
- Giddings ve Lewis General Electric
- Morey General Dynamics
- Cincinnati EMI (British)

EMI tarafından yapılan denetim ünitleri analog türde olup diğerleri sayısaldı. Daha sonraları analog tasarım yetersiz bulundu ve sayısal tiplerle değiştirildi. Riski azaltmak için tasarlanan bölünmüş kontratlar elverişli sistemler yarattı ama bununla beraber farklı tasarımlı NC denetim sistemleri ve NC programlarında farklı format ve kod kullanılması sonucunu doğurdu ki bu da NC programlarının taşınabilirliğini (aynı parçayı üretmek için farklı marka NC tezgahlarında aynı NC programının kullanılması) engelledi.

Bu NC tezgahlar 1958 ile 1960 arasında bir çok değişik uçak yapım fabrikalarında hizmete alındı. Kullanıcılara tamamen yeni idiler ve teknisyenler ile mühendisler tarafından geleneksel tezgahlara uygulananlardan farklı davranışlar gerektiriyorlardı. Bununla beraber, NC tezgahların monte edildiği fabrikalardaki teknik personel bu durumun farkında değildi. Üstelik bu aşamada endüstride programlama deneyimi olan kişi sayısı çok azdı. Sonuçta bu makinaların çoğu bozuldu ve hatta bazıları parçalandı. Yangına körükle gidercesine, tasarımlarının tam denenmemiş olmasından ve elektronik sistemlerin bugünkü kadar güvenilir olmamasından dolayı, bu NC denetleyicilerinde sık sık problemler oluştu. Bu önceden görülemeyen problemlerden dolayı cesareti kırılan bir çok kullanıcı NC tezgahların kullanımına devam etmek istemedi. Hava Kuvvetlerinin projeye devam etmek için kullanıcıları ikna etmesi oldukça çaba gerektirdi.

Zorluklar NC tezgah yapımcıları tarafında, NC denetim sistemlerinin tasarımlarını geliştirerek, kullanıcı tarafında da programcıları, operatörleri ve bakım teknisyenleri eğiterek yavaş yavaş yenildi. Problemler sonunda 1961 ile 1962 arasında kontrol altına alındı. Yeni teknolojinin üstünlüklerinden tatmin olan uzay ve havacılık firmaları artık kendileri ödeyerek yeni NC tezgahları yapmaya veya sipariş etmeye başladılar.

Bu arada daha basit NC tezgahları, ismiyle noktadan-noktaya NC delme tezgahları programlanmalarının kolay olmasından ve kontrol birimlerinin basitliğinden dolayı çok yaygın bir tanınmışlığa ulaştı.

İlk NC tezgah, MIT tarafından düşey milli bir Cincinnati Hydrotel freze tezgahının yeniden düzenlenmesi ve içine NC denetim birimi eklenmesi ile geliştirildi. Tezgah geleneksel tasarımlı idi ve NC denetim biriminde kullanılan lambalar onu ağır ve güvenilir kılıyordu. O sadece NC tezgahların olabileceğini kanıtladı. NC tezgah denetim biriminin tasarımının havacılık üretiminin gerektirdiği işleme hassasiyetini elde edebilmek için belirgin olarak geliştirilmesi zorunlu idi.

Tarihi olarak NC uygulamalarının başarısı iki ana etkene dayanır: NC'li tezgah denetim birimlerinin geliştirilmesi ve yazılım programlama araçlarının geliştirilmesi. Çağdaş bir NC metal-kesme tezgahının yapısı, düzenlenmesi ve tasarımı geleneksel takım tezgahları ile karşılaştırıldığında oldukça değiştirilmiştir. NC tezgahının ilerleme sistemi, sistemin en hayati önem taşıyan parçasıdır çünkü o tezgahın çevreleme ve konumlama hassasiyetini saptar. NC'li tezgahlarda kullanılan ilerleme sistemlerindeki gereksinimler yüksek hassasiyet ve hızlı cevaptır. Kızaklar ve hareketli parçalar arasında kullanılan sürtünmesiz tasarımlı yuvarlanan elemanlı kızak yolları sürtünmeyi ve geleneksel tasarımlarda görülen stick-slip* etkisini azaltmak için kullanılır^{3,4}. Temel eleman olan sürtünmesiz tasarımlı dönüşümlü bilyalı vida bilinen Trapez vidayı değiştirmiştir. Süren ve sürülen elemanlar arasındaki boşluktan doğan konumlama hatasını azaltmak için boşluksuz sürüş mekanizmaları geliştirildi. Geleneksel tezgahlardaki merkezi sürüş tasarımının yerine ayrı ayrı doğru akım motorları değişik eksenlerin hareketini sağlamak için kullanıldı.

Kapalı devre kontrolün gerçekleştirilebilmesi için gerekli olduklarından hem açışal hem de doğrusal hareketler^{5,6} için yüksek hassasiyetli konum ölçme ve geri besleme sistemleri geliştirildi ve NC'li tezgahlarda kullanıldı.

Klasik X, Y ve Z eksenlerine ek olarak doğrusal ve dönme hareketlerini de kapsayan ek hareketler karmaşık çevrelerin işlenebilmesini için NC tezgahlara eklendi. Bundan başka bir parçayı bir bağlayışta farklı kesiciler ile işleyebilmek için otomatik takım seçici ve değiştirme sistemleri NC tezgahlara eklendi. NC işleme merkezi olarak tanımlanan yeni bir tezgah 1950'lerin sonuna doğru ortaya çıktı.⁷ Bu tezgah frezeleme, tornalama, delik işleme ve delme gibi bir çok geleneksel tezgahın görevlerini birleştirdi.

Bu arada, NC'li takım tezgahlarının yapısal düzenlemeleri de belirgin değişikliklere uğradı. Tezgah gövdesinin daha dirençli olmasına, araba ve döner başlığın daha iyi desteklenmesine ve talaşın kolayca atılabilmesine bu faktörler yüksek hassasiyetli işleme ve yüksek oranda metal kesebilme için önemli etkenlerden oldukları için önem verildi. Hem çok yönlülük hem de basitlik için 1960'ların sonundan bu yana NC'li tezgahların tasarımında eğilim modüler tasarım kavramı* olmuştur.⁸

NC tezgahların geliştirilmesinde başka bir önemli görünümde NC denetleyici olarak adlandırılan, NC kontrol sistemi ile ilgilidir. NC denetleyiciler iki tipe ayrılır: açık ve kapalı devre kontrol tipleri. İlki konum hatalarının giderilmesine imkan tanımaz ve ve konumsal ve noktadan-noktaya düşük hassasiyetli denetim gerektiren işlerde kullanılır. En çağdaş NC'li tezgahlar ise konum ölçme biriminden geri

* Stick-slip terimi bir makina elemanının diğer makina elemanı üzerinde yavaşça kayma hareketi esnasında statik ve dinamik sürtünmelerin birleşik etkisi ile ortaya çıkan ve istenmeyen küçük ve kesikli takılma ve sıçrama hareketlerini tanımlamak için kullanılır.

* Modüler tasarım kavramına bağlı olarak, bir tezgah birkaç temel fonksiyonel birimden meydana gelir. Eğer bu birimlerin tasarımı standartlaştırılırsa, tezgahın tasarımı ve üretimi çok fazla basitleştirilir.

beslenen komün hatalarını gideren ve böylelikle konumsal ve çevreleme izleme kontrolü için gerekli hassasiyet ve tekrarlanabilirliği garanti edebilen kapalı devre kontrol ilkesine dayalıdır.

İlk NC denetleyicisinde lambalar ve elektrikli röleler kullanılmıştır; tezgah üzerindeki denetimi bir hidrolik servo mekanizma tarafından gerçekleştirilmiştir. Bu tür bir denetim sistemi güvenilir ve hassas değildir. Elektronik teknolojisindeki gelişme ile, ikinci ve üçüncü nesil NC denetleyiciler tek transistörler ve entegre kartlar kullanılan sayısal devrelerle yapılmışlardır. Bu denetleyiciler NC programının özel kodlarla yazılmasını, delikli kağıt şeritte saklamasını ve teyp okuyucu tarafından denetleyiciyi beslemesini gerektirmiştir. Üstelik temel denetim görevleri donanım (elektronik devre gibi) tarafından gerçekleştirilmiştir, bu yüzden yüksek maliyetli NC denetleyiciler ortaya çıkmış ve onların daha gelişmiş fonksiyonları gerçekleştirmekten alıkoymuştur. Bilgisayar teknolojisinin gelişmesi bilgisayar donanımının maliyetinin devamlı olarak düşmesi sonucunu ortaya getirince 1960'ların sonuna doğru ROM, sadece okunabilir bellek teknolojisi NC denetleyicilere uygulanabilmiştir. İşleme emirlerinin sıraları ROM'ların içine saklanabilmiş ve ulaşılmış ve MCU- tezgah denetim birimi tarafından işletilebilmiştir. Sonuç olarak bir NC denetleyici daha fazla donanım eklenmesine bağlı kalmadan daha fazla fonksiyona sahip olabilmıştır; gereken şey sadece saklanmış bir programdır. NC denetleyici daha karmaşık ama daha ucuz hale gelmiştir. Bilgisayarlı sayısal denetim olarak adlandırılan teknik de mikroişlemcilerin ve bilgisayarların boyutlarında küçülme devam ettikçe 1970'lerin sonunda özel bir bilgisayarı bir NC denetleyicinin içine yerleştirmek mümkün olmuştur. Artık kağıt veya manyetik şerit CNC kontrol birimi içerisinde program saklamak için gerekli tek araç değildir. Bir program kontrol biriminin bellek biriminde saklanabilir veya ayrı bir bilgisayardan alınabilir. Aynı şekilde CNC denetleyicinin içinde saklı program değiştirilebilir, bu daha evvelki NC denetleyicilerin hiç bir zaman sahip olamadığı bir rahatlıktır. CNC denetleyicisi aynı zamanda tezgahın durumunun tanımlanmasını ve değişik giriş - çıkış araçları ve bilgisayarlarla haberleşmenin de kolaylaşmasını sağlamıştır.

Bir CNC denetleyicisi ile bilgisayar arasından iletişimi sağlamak için iki yaklaşım geliştirildi. DNC, dağıtılmış sayısal denetim ile tamamlanmış bir parça işleme programı bir bilgisayardan yollanabilir ve çalıştırılmadan önce CNC denetleyicisi içerisinde saklanabilir. Aynı şekilde CNC denetleyiciye bir yandan çalışırken NC programı deyim deyimde yollanabilir; bir NC programını parça işlenirken gerçek zamanda gönderme metodu doğrudan sayısal denetim olarak yine DNC, adlandırılır. Bu iki yaklaşım arasındaki fark, dağıtılmış sayısal denetim sisteminde program boyutunun NC denetleyici belleğinin kapasitesi ile sınırlı olmasıdır. Doğrudan sayısal denetim ile NC'li tezgahın çalışması bilgisayar tarafından gönderilen sinyallere bağlıdır. Genellikle merkezi bir bilgisayar bir çok NC'li tezgahu

kontrol etmek için kullanılır. Bu nedenle NC'li tezgahın çalışması kontrol bilgisayarının güvenilir çalışmasına bağlıdır.

Bügüne kadar NC teknolojisi tarafından kaydedilen başarı, NC programlamayı kolaylaştırmak için geliştirilen yazılımlar geliştirilmemiş olsaydı imkansız olacaktı. Yukarıdaki açıklamalardan görüleceği gibi bir işleme operasyonundaki takım konumlarının birbirini izleyen konumları ile ilgili kesin veriler bir NC programında verilmelidir. Karmaşık parçalar için takım konum verilerini elle hesaplamak mümkün değildir. Hatta NC kesici izleri elle hesaplanabilen parçalar için harcanan programlama zamanı oldukça büyüktür. Programlamadaki diğer bir güçlükte verilerin farklı NC denetleyicileri tarafından gereken kodlara çevrilmesinde doğar.

1955'de MIT tarafından prototip bir NC programlama sistemi geliştirildi ve Whirlwind bilgisayar üzerinde bilgisayarların programlamaya yardımcı olabileceklerini göstermek üzere test edildi. 1957'de Uzay ve Havacılık Endüstrileri Birliği organizasyonunun üyeleri, bütün NC sistemleri için şerit hazırlanmasına yardımcı olacak bir programın hazırlanması için ortak bir çalışmaya başladılar. Bu programın geliştirilmesi MIT'in koordinasyonu altında bu uzay ve havacılık firmalarının araştırmacı matematikçileri tarafından yürütüldü. 1958'lerin sonunda tamamlanan bu bilgisayar programı, APT (Takımların otomatik programlanması) olarak adlandırıldı. Bu IBM bilgisayar sistemleri üzerinde kullanılmak üzere tasarlanmıştı. Bu ilk sürüm bir çok hataya ve eksikliklere sahipti ve deneme çabaları, hatalardan arındırılması ve iyileştirilmesi 1960 Haziran'ına kadar tamamlanamadı. Daha sonra çok iyi bilinen APT III programı veya işlemcisi iyileştirildi. 1960'ların başında endüstri de en çok kullanılan yazılım olduğundan APT aynı zamanda bu uzay ve havacılık firmaların ana çatı bilgisayarlarında yüklenmiş en çok kullanılan yazılımdı. (%30 üzerinde kullanım).²

APT geliştirilmesi esnasında alınan bir önemli karar da şu idi; Hava Kuvvetleri tarafından desteklenen dört NC sisteminin hepsi tarafından da kullanılacak şekilde tasarlanmak zorunda idi. Böylece takım konumlarının ve istenen işlem sıralarının tanımlanması ve takım konum verisi olarak (CLDATA) bilinen APT işlemcisinden alınan çıkış, NC sisteminden bağımsız olarak standard formatta olmak zorunda idi. Daha sonra başka bir bilgisayar programı tarafından, sonişlemci postprocessor olarak tanımlanan başka bir bilgisayar programı tarafından kullanılan NC'li tezgahın belirli NC kodlarına çevrilmesi gerekiyordu.

1961'de kazanç gözetmeyen bir grup olan APT Long Range Program, APT yazılım paketini geliştirmek ve desteklemek için kuruldu. Görev Illinois Institute of Technology Research Institute verildi. Program daha sonra bir çok Avrupa ülkesinin ve Japonya'nın katılımı ile uluslararası bir konu haline geldi.⁹

APT'nin ilk sürümü temel olarak üç eksenli veya çok eksenli çevresel işleme için tasarlanmıştı. İngilizceye benzer bir şekilde yazılan, APT dili olarak tanımlanan parça işleme programı, parçanın geometrisini, kesici izlerini ve işleme

özelliklerini tanımlar. Noktadan noktaya hareketleri veya iki boyutlu işlemin çevrim hareketlerini içeren bir çok basit işleme operasyonlarında, dil kullanıcı için pek uygun değildir. APT'nin küçük kullanıcılar için diğer bir sakıncası da 1960'lar da APT işletmek için çok pahalı olan büyük bir ana çatı bilgisayara ihtiyaç göstermesidir. Bundan dolayı bir çok NC programlama ve işleme programları hem Birleşik Devletler'de hem de Avrupa'da APT programını basitleştirmek ve bilgisayar gereksinimini azaltmak ve APT'de var olmayan bazı fonksiyonları eklemek için geliştirildi. Bu programların bir çoğu APT10 üzerine veya APT'ye benzer ilkelere dayalıdır. Bunlara örnek olarak ADAPT, EXAPT, IFAPT, MINIAPT, NELAPT ve COMPACT II (ki APT'den türetilmemiştir.) dahildir. Sadece APT ve COMPACT II bu ülkede genel kullanımdadır. Bugün APT Dünya çapında kabul görmüş tek NC programlama dilidir ve Birleşik Devletler'de 1974'ten bu yana standardlaştırılmıştır.¹¹ Halihazırda APT, IBM firması tarafından desteklenmektedir.

Yüksek seviyeli NC programlama dillerinin başarılı kullanımı aynı şekilde CLDATA'nın sonişlenmesine de dayalıdır. Sonişlemci genellikle kullanıcı tarafından NC'li tezgah-denetleyici sisteminin özelliklerine ve kullanıcının bilgisayar sistemine göre tasarlanır. Sonişlemci belli bir büyüklükte karmaşık programlar oldukları için tasarlanmaları normal kullanıcılar için fazla büyük bir problemdir; bu da bir çok kullanıcının otomatik programlama sistemini başarılı bir şekilde kullanmasını engellemektedir.

Gelişmiş bir NC programlama dili olarak, APT NC programlama ve şerit hazırlama işini gerçekten basitleştirdi. APT geometrik tanımlama deyimleri ile tanımlanmış geometrik öğeler dayanarak bir NC kesici izi oluşturulabilir. 1960'lardaki, bilgisayar destekli tasarım (CAD) teknolojisinin hızlı gelişmesi tasarımcıların mühendislik çizimlerini CRT katod ışıklı tüp CRT üzerinde hazırlamalarını ve geometrik modeli bilgisayar üzerinde yapmalarını mümkün kıldı. Bilgisayarda yer alan CAD sistemi tarafından tanımlanan bir geometrik öğe, verilerden oluşan bir grup ve geometrik öğeyi tanımlayan matematiksel denklem veya fonksiyonların parametreleri tarafından temsil edilir. Bir başka deyişle, böyle bir veri bilgisayarda temsil edilen böyle bir veri parçanın CAD modeli olarak tanımlanır. Bir CAD modelinin verileri daha ileri aşamada NC programlama yazılımlarının yardımı ile NC kesici izlerini tanımlamak için de kullanılabilir. Böylece eğer bir CAD sistemi CAD modeline dayanarak NC kesici izlerini saptayabilecek işlevlere sahipse, herhangi bir kimse doğrudan CRT ekran üzerine çizilmiş resmi kullanarak NC tezgah işlemlerini tanımlayabilir. Böyle bir sistem genellikle CAD/CAM, bilgisayar destekli tasarım/bilgisayar destekli üretim sistemi olarak adlandırılır. CAD/CAM sistemleri 1980'den önce yüksek maliyetlerinden ve güvenilir olmamalarından dolayı yaygın olarak kullanılmadı. Bugünlerde oldukça fazla sayıda genel ve NC amaçlı CAD/CAM sistemleri kullanımdadır.

NC teknolojisi başta metal çevreleme işlemlerini kontrol etmek için geliştirilmişti. Bugün halihazırda onun uygulaması bu ufku çoktan aşmıştır. Tornalama, frezeleme, delme, taşlama ve elektroerezyon işlemeyi kapsayan metal kesme uygulamalarının yanında NC teknolojisi aynı zamanda kaynak, alevle kesme, metal şekillendirme (saç levha işleme, haddeme ve dövme gibi çok geniş uygulama alanları olan) muayene ve ölçme işlemlerinde kullanılmaktadır. NC metal kesmeden başka ağaç işleme, tekstil, plastik ve elektronik gibi endüstrilerde de kullanılmaktadır. Robotik bugün aynı programlama ve kontrol felsefelerinin uygulanmasından dolayı NC'nin en yakın önemli uygulamalarından biri olarak düşünülebilir.

1.2 NC TEKNOLOJİSİNİN BUGÜNKÜ DURUMU

NC teknolojisinin tarihçesi ABD'nin NC'nin bütün yönlerine en çok katkıda bulunan millet olduğunu göstermektedir. 1970'lerin ilk yıllarında ABD'de yapılan NC'li takım tezgahlarının sayısı herhangi bir başka ülkedekilerden çok çok fazla idi. Ondan sonra Birleşik Devletlerdeki NC'li takım tezgahu üretimi Japon ürünlerinin düşük fiyat ve elektronik ürünlerinin güvenilirliğinin üstünlüğünden dolayı Japonların yavaş yavaş altına düşmeye başladı. 1985 ABD NC'li takım tezgahu ihtiyacının %53.3'ünü ithal ediyordu.¹³Başka bir istatistik¹⁴ 1980 ile 1985 arasında ABD'deki sadece %18'lik oran ile karşılaştırıldığında, Japonya'da sunulan tezgahların %55'inin CNC tezgahlar olduğunu gösteriyordu. NC'li takım tezgahları ABD metal işleme endüstrisinin tezgahlarının toplamının küçük bir bölümünü* oluşturmaktadır; bunula beraber artış hızı belirgindir. 1985'te yıllık NC'li takım tezgahu tüketimini, dolar olarak toplam ABD tezgah ihtiyacının %43.8'ine ulaşmıştı. NC takım tezgahları geleneksellerden daha yüksek üretkenlikte olduğu için, üretimdeki etkileri halihazırdaki makina birimi yüzde değerinde yansıtılandan daha yüksekti.

30 yıldan fazla bir gelişmeden sonra NC'li takım tezgahlarının tasarımı ve performansı yeni bir seviyeye ulaştı. Bugün yapılan NC'li tezgahların çoğu çok yeni sayıda unsura sahip CNC denetleyicileri ile donatıldı. Bütün CNC denetleyicileri, operatörlerin NC programlarını daha kolayca gözden geçirip düzeltmelerini mümkün kılan CRT karakter ekranları ile donatıldılar. Denetleyiciler giderek daha fazla sayıda kesici yarıçapının telafisini sağlayan, işleme çevrimlerinin daha kolay elle programlanmasını sağlayan ve simetrik kesici izleri (ayna görüntü işlevi) için programlara sahip oldular. Basit karar ve hesaplama deyimleri değişken (veya MACRO) programlama kapasitelerine sahip denetleyicilerin NC programları içine dahil edildi. Mazatrol CAM M-2 CNC denetleyicisi gibi en yeni CNC denetleyicileri aynı zamanda grafik ekran ve yazılımlara sahip olup kalıp işlemesi ve bazı

* 1983'te bu rakkam %4.7 idi.¹³

üç boyutlu çevreleme işlemlerinin elle programlanmasını çok basitleştirmektedir. Grafik ekran kesici izlerini gösterebilmekte ve böylece NC'li tezgah gerçekten çalıştırılmadan önce bir programın test edilmesine izin vermektedir. Robotlarla, ölçme cihazları ve bilgisayarlarla iletişim CNC denetleyicisi üzerinden gerçekleştirilebilir. NC kontrol tasarımında yeni bir yaklaşımda kesme kuvveti, takım sıcaklığı, motor momenti ve takım aşınması gibi konum ve hız geri besleme döngüleri tarafından kontrol edilemeyen bu karakteristik değerleri hissedebilen ve hesaplayabilen adaptif kontrol sistemlerinin kullanılmasıdır ve böyle bir adaptif sistem NC komutlarını optimum metal kesme veya güvenlik koşullarını koruyacak şekilde değiştirebilir.

Çağdaş CNC tezgahları tek bir aksel yönünde çok yüksek konumlama hassasiyetlerine (tam strokta ± 0.008 ile ± 0.015 mm ve ± 0.0003 ve ± 0.0006 inç gibi) ve konumlama tekrarlanmasında (± 0.001 ile ± 0.002 mm gibi veya ± 0.0004 ile ± 0.00008 inç gibi)¹⁶ çok yüksek hassasiyetlere ulaşmışlardır. Çoklu eksenlerde konumlama hassasiyeti aynı şekilde kızakların form hassasiyetine ve değişik eksenler arasındaki açısal konum ilişkilerine bağlı olup, nispeten düşüktür. En yüksek devir sayıları 4000 ile 8000 devir/dakika arasındadır; bazı CNC tezgahlarda tasarlanmış en büyük devir sayısı 75000 devir/dakika⁷ kadar yüksek olabilmektedir. Kesme ve hızlı ilerleme değerleri 5000 mm/dak (veya 197 in/dak.) ile 15,000 mm/dak (veya 590 in/dak) gibi hızlara ulaşmaktadır. Bundan dolayı yüksek miktarlarda malzeme kesme miktarlarına ulaşabilmektedir.

Bir çeşit otomatik parça yükleme ve boşaltma sistemi olan palet sistemi son işleme merkezi tasarımlarının ana unsuru haline gelmiştir. Hatta bazı CNC tezgahları üretkenliğini artırmak için parçaları yüklemek ve boşaltmak için tahsis edilmiş robotlar veya manipültörler kullanılır.

Elektronik yüzey-izleyen duyarlı uçlar gibi otomatik ölçme cihazları NC tezgahlarda parçanın konumunu saptamak ve tezgah koordinat sisteminin başlangıcını ayarlamak için kullanılmaktadır.

Bir tezgah işleminde hem kesici yolunu hem de tezgah işlemlerini kontrol için kullanılan teknoloji olan sayısal denetim, büyüklüğünü NC'li tezgahlar bilgisayar kontrolü altında malzeme iletmeye ve ölçme araçları ile beraber bir (FMS), örnek üretim sistemi birlikte kullanıldığında göstermektedir. FMS, merkezi bir denetim bilgisayarının kontrolü altında parça tasarım ve üretim gereksinimlerine göre tasarım sınırları içerisinde kendi üretim düzenini ayarlayan bir otomatik kontrol sistemidir. Bunun çağdaş üretim otomasyonunda kritik bir yeni alan olduğunu öngören Japonlar mekatronik olarak tanımlanan bu alana geçmiş on yılda çok büyük önem verdiler ve yatırım yaptılar. Sonuçta yeni teknolojinin önderliğinde diğer ülkelere göre çok belirgin bir noktaya gelmeyi başardılar.¹⁴

NC programlama yazılımları konusunda, APT işlemcisinin son sürümlerine daha çok işlev eklendi. En son sunulan IBM APT-AC Nümerik Kontrol İşlemcisi*

* Kitapta Bölüm 8.5.'e bakınız.

sınırlanmış geometrik öğelerin, tornalama çevrimlerinin ve temel geometrik elemanların bir kontur kümesi olarak tanımlanmasına izin vermektedir. Aynı zamanda tezgah işleminin bilgisayarın grafik sistemi ile iletişimli olarak programlanmasına izin vermektedir.

Halihazırda çok sayıda NC amaçlı CAD/CAM sistemleri değişik kapasitelerde mevcuttur. Bunlardan bazıları aynı zamanda sonişlemci tasarımı için kullanılan araçları içermektedir.

NC programlama sistemlerinden elde edilen çıktı (örneğin CLDATA) sonişlemci tasarımı ve farklı sistemler arasındaki iletişimi basitleştirmek için standardlaştırılmıştır. Sonişlemci, CAD/CAM sistemlerinin NC programlarının oluşturulmasına ve farklı NC sistemleri arasındaki iletişimin başarılı bir şekilde kullanılmasına önemli bir engel olmaya devam etmektedir. Aynı zamanda sonişlemci tasarım araçları bazı bilgisayar destekli NC programlama sistemlerinin kullanışlı bir unsuru haline gelmiştir. Sonişlemci tasarımındaki en yeni eğilim kullanıcıların sadece bazı ard arda gelen soruları cevaplandırarak bir sonişlemciyi tasarlamasına izin veren gelişmiş programların NC programlama sisteminin içinde sunulmasıdır.

NC programının otomatik olarak doğrulanması hala çözülmemiş bir problemdir. Welch¹⁷ tarafından 1980'de yapılan ihtiyatlı bir tahmine göre, örnek bir uzay ve havacılık firması her 75 CNC tezgah için NC programlarının doğrulanmasına yaklaşık bir milyon dolardan fazla harcamaktadır. Genel olarak, bir NC program doğrulama sistemi program sözdizimini kontrol etmeli, kesici konumlarını ve oluşturulmuş profili tasarlanmış parça profiline göre kontrol etmeli ve eğer varsa tezgahın kesici takımını da kapsayan bağlama aparatı, parça malzemesi ve tezgah tablası gibi hareketli parçaları arasındaki muhtemel çarpışmaları tesbit edebilmelidir. Parça programlarının doğrulanması gayesiyle sistem geliştirme çabaları için Dünya çapında oldukça büyük bir çaba harcanmaktadır.¹⁸ Doğrulama sistemlerinin çoğu kesici yolunu temel olarak iki veya iki buçuk boyutlu olarak gösteren bilgisayarlı grafik sistemlerini kullanmaktadır. (*) Hala parça-takım tezgahı-kontrol biriminin dinamik performansını taklit etme kapasitesine sahip yeterli bir sistem yoktur. Genellikle parça işlenmeden önce tezgah üzerinde bir deneme çalışması yapılmalıdır.

NC programının CAD modeline dayanarak doğrudan otomatik olarak üretilmesi ve takımları, tezgahları, bağlama aparatlarını, malzemeleri, ve üretim takvimlerini kapsayan bir veri tabanının doğrudan kullanılması NC programlama teknolojisinde önemli bir gelişme eğilimini temsil etmektedir.

1.3 NC TEKNOLOJİSİNİN TASARIM VE ÜRETİM ÜZERİNDEKİ ETKİSİ

NC teknolojisinin tarihi, sayısal denetimi, imalatın geleneksel şartlarından çok çeşitli ürünlere ve hızlı değişen ürün tasarımına olan ihtiyacına cevap vermek

* 2 veya 1/2 boyutlu kesici yolu, kesicinin işleme derinliğinin değişmesinde üçüncü boyuttaki hareketin dışında, iki eksenle aynı anda hareket etmesidir.

üzere gelişmiş bir otomasyon aracı olarak belirtmektedir. NC, geleneksel yüksek hacimli katı otomasyon ile bilgisayar kontrollu esnek olan arasındaki bir köprü olarak görülebilir. Milner ve Vasiliou'nun belirttiği gibi "son yirmi yılda" "çok az yeni mühendislik ve üretim işlemleri bu kadar çok merak uyandırmış ve endüstrinin bir çok kolunda bu kadar çok değişikliğe neden olmuştur".¹⁹

NC bir üretim yöntemi olarak, ilk defa bir insanın özelliklerine veya kullanımının yeteneklerine bağlı olmadan herhangi bir şekilde sahip mekanik parçaların üretilmesini mümkün kılmıştır. Tasarımcılar daha önce yapılması ekonomik olmayan veya üretilemez olarak tanımlanan biçimleri, yüzeyleri ve profilleri tasarımlarında artık belirtebilirler. Yüksek konumlama hassasiyetinden, çok eksenli hareketlerden ve çoklu görevlerinden dolayı çağdaş CNC tezgahlarla, birbirlerine göre çok kesin konumlu ve yönlendirmeli gereksinimlerle bir araya getirilmiş geometrik elemanlardan oluşmuş parçaları kolayca işlemek mümkün olmuştur.

NC işleme ve CAD/CAM sistemleri parçaların ve kesici yollarının matematiksel formüllere ve tek referans koordinat sistemine bağlı olarak tanımlanması özelliği ile karakterize edilebilirler. Bu tasarım resimleri üzerinde boyutların ve toleransların tanımlanma biçimini etkiler. Parça unsurlarının temel koordinat sistemine bağlı olarak tanımlanması, yüzeylerin ve profillerin matematiksel olarak tarif edilmesi ve iki taraflı toleranslandırma CAD/CAM ve NC sistemi ile tasarlanacak ve üretilecek parçaların boyutlandırılmasında ve toleranslandırılmasında önerilen kılavuz elemanlar arasındadırlar.

NC'nin bir ürün üzerinde etkisi geleneksel olarak tasarlanan parçaların öncelikle fonksiyonel gereksinimlere sonra da ne kadar kolay üretilebildiğine bağlı olduğundan sadece belirli bir bölümle sınırlıdır. Bununla beraber NC'nin üretim üzerinde etkisi bir manifold gibidir. Öncelikle o tasarlanan veya planlanan işlemin veya üretim yönteminin yolunu değiştirir. Örnek olarak geleneksel işleme için hazırlanmış bir üretim planı tezgahlarla, takımlarla, bağlama aparatları ile, zaman oranları ile, kullanılan işlemlerin sıraları ile ve ilerleme hızları, kesme hızları gibi gerekli işleme özelliklerini kapsayan bilgileri sunar. Bununla beraber tornalama veya frezeleme gibi tek tek işlemlerin çalıştırılması genellikle operatör tarafından deneme-yanılma yöntemine dayanan verilerle belirlenir. İnsan müdahalesi en aza indirilmiş NC ve çağdaş bilgisayarla bütünlüklü üretim (CIM) yöntemlerinde, bir işlem planı veya ilgili işlemlerin tümü bir bilgisayar tarafından çalıştırılabilir şekilde program veya yordam olarak ve NC'li tezgah tarafından tamamen takip edilebilecek şekilde tasarlanabilir. Sonuçta bir işlem planlayıcı her bir işlem hakkında çok derin bilgiye sahip olmalıdır ve NC programlayıcı bir operatör veya teknisyen gibi işleme operasyonunun her detayını bilmelidir. İkinci olarak, NC işlemin nasıl yerine getirildiğini de değiştirir. İşlemler artık operatör tarafından değil NC programı tarafından kontrol edilmektedir. Bir operatörün becerisi bilginin işlenmesi (işleme adımlarının NC programında tanımlanması) ile yer değiştirmiş-

ti; sonuçta becerili tezgah operatörlerine olan ihtiyaç azalmaktadır. Hazırlık işlemlerinin çoğu programcı tarafından yapılacağından tezgahattan alınabilir ve işlem için gerekli ayarlama zamanı çok büyük oranda azaltılabilir. Üçüncü olarak NC bir işlemin veya operasyonun nasıl kontrol edildiğini değiştirir. Bir işlem veya operasyon üzerindeki elle kontrol NC denetim birimi veya bilgisayarın kontrolü ile yer değiştirmiştir.

NC'nin imalat sanayii üzerindeki darbesi çok derindir. O otomasyonun felsefesini, bir işlemin nasıl tasarlandığını, yerine getirildiğini ve kontrol edildiğini ve imalat sanayiiindeki iş gücünün yapısını değiştirmiştir. Şimdi insansız üretimi gerçekleştirmek mümkündür; böylece esnek otomasyon bilginin bilgisayarlar tarafından çok hızlı işlenmesi ile tasarımdaki ve üretimdeki değişikliklere kolayca uyarlatabilen otomasyon çeşididir başarılıdır.

NC tarihinin bir hatırlatması ve çok sayıdaki bilgisayar destekli üretim üzerine hazırlanmış makalelerin belirttiği bir görüntü olarak otomasyona bir yaklaşım olan NC geçmişte üretimde devrim yapmıştır ve bilginin hızlı işlenmesi ve paylaşılması ile karakterize edilebilen çağdaş bilgisayar destekli üretimin önemli bir kurucusudur.

SORULAR

- 1.1 Nümerik kontrol kendi terimlerinle tanımlamasını veriniz.
- 1.2 Geleneksel otomasyon yöntemleri ile karşılaştırıldığında nümerik kontrol, esnek veya programlanabilen otomasyona bir yaklaşım olarak tanımlanabilir. Bu iki tip otomasyonu karşılaştırınız ve nümerik kontrolün kullanılması sayesinde otomasyonun neden esnek hale geldiğini açıklayınız.
- 1.3 NC tezgahlarının ve teknolojisinin ortaya çıkmasına ve gelişmesine hangi etkenler neden olmuştur?
- 1.4 Etkin bir NC sistemi oluşturmak için ABD Hava Kuvvetleri, farklı tasarımlara sahip rakip NC sistemlerinin yapılmasına karar aldı. Bu kararın NC ve CIM teknolojisine en derin etkisi nedir?
- 1.5 Çağdaş NC'li tezgahların gerçekleştirilmesi için takım tezgahları tasarımındaki önde gelen gelişmeleri sıralayınız.
- 1.6 NC programlama araçlarını sağlayan yazılım, NC teknolojisinin ve tezgahlarının başarılı uygulanmasında neden bu kadar önemli idi?
- 1.7 NC ile CNC tezgahları arasındaki temel fark nedir? NC tezgahlarının yerini neden CNC tezgahları almıştır?
- 1.8 "Dağıtılmış Nümerik Kontrol" ve "Direkt Nümerik Kontrol" ve aralarındaki farkları açıklayınız.

- 1.9 Neden APT gibi, gelişmiş bir programlama dili gerekli olmuştur? NC programlamaya sağladığı yardım nedir?
- 1.10 NC programlamadaki üç yaklaşım nedir? Herbirini diğerleri ile karşılaştırınız.
- 1.11 NC teknolojisinin geliştirilmesindeki bugünkü eğilim nedir?
- 1.12 NC'nin CAM sistemlerinin ve CAM teknolojisinin geliştirilmesinde oynadığı rol nedir?
- 1.13 Sonişlemci varlığının CIM sisteminin gerçekleştirilmesinde neden olduğu problem nedir? Bunu çözmek için getirilen yaklaşımlar nelerdir?
- 1.14 NC programının doğrulanması nedir? Üretim esaslı NC teknolojisinde neden önemlidir?
- 1.15 NC'nin ürün tasarımındaki ve üretimindeki etkisi nedir?

Bölüm 2

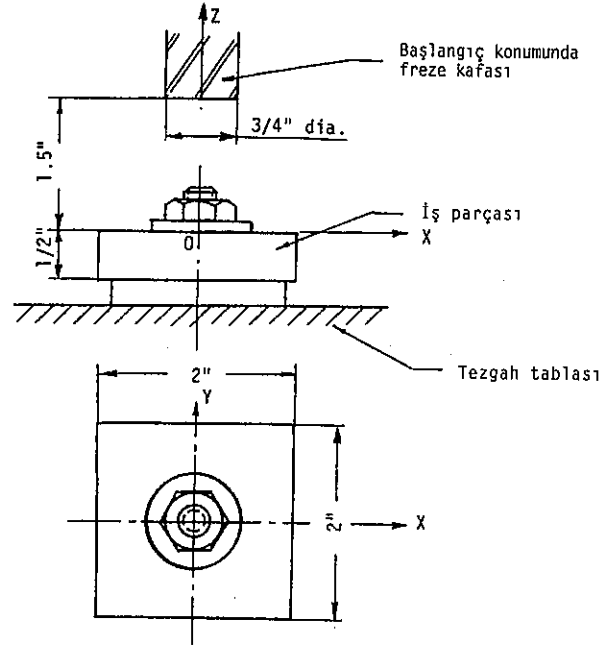
Tezgah İşlemlerinin Elle NC Programlanmasına Giriş

NC'li takım tezgahının çalıştırılması NC kodunda yazılmış, belli bir parçayı işlemek için NC'li tezgah tarafından gerçekleştirilmesi gereken kesici hareketlerini ve yürütülmesi gereken işlemleri tanımlayan deyim veya blok dizilerinden oluşan NC programı olarak adlandırılan program tarafından kontrol edilir. Klasik metal işleme takım tezgahlarında, genellikle bir tezgah işlem planı tanımlanır, parçayı gereken boyutlarda ve toleranslarda üretmek için operatör tarafından takip edilir. Bir NC programı tezgah işlem planının İngilizceden NC denetleyicisi tarafından anlaşılabilir NC kodlarına çevrisidir. Program önce okunur ve sonra NC tezgah-denetleyici tarafından gerçekleştirilir. Uluslararası Standard ISO 6983/1 gibi NC kodu standartları olmasına rağmen farklı üreticiler tarafından yapılmış NC denetleyiciler farklı NC kodlarını kullanırlar. Bu yüzden NC programı farklı tasarımlardaki denetleyiciler üzerinde taşınabilir değildir. Bununla birlikte, NC kod formatları ve çağdaş CNC denetleyicilerinin programlanması birbirine çok benzerdir. Bu yüzden elle NC programlamanın genel prensiplerini ve karakteristiklerini belli bir CNC denetleyicilerinin programlanmasını çalışarak açıklamak mümkündür. Bu kitapta elle NC programlamanın temel ilkelerini açıklamak için freze tezgahı için FANUC 6MB, torna için FANUC 6T denetleyicilerini kullanacağız, bunlar Dünya üzerinde en yaygın kullanılan denetleyicilerdir.

Şekil 2-1 kare bir parçanın çevresini işlemek için kullanılan, işleme planını da içeren basit bir NC programını göstermektedir. Bu örnekten görüleceği gibi bir NC programı aşağıdakilerden oluşur:

1. Program numarasının beyan edilmesi.
2. Koordinat sisteminin orijinini ve takım konumunu veya yolunu programda tanımlamak için kullanılan koordinat sisteminin türünü belirten deyimler.
3. NC'li tezgah tarafından gerçekleştirilecek takım hareketleri dışındaki soğutma sıvısını ve iş milini açma ve kapatma, devir sayısını ve ilerleme hızını ayarlama ve takımları değiştiren veya seçen işlemleri tanımlayan deyimler.

4. Takımın konumunu ve hareketlerini tanımlayan deyimler.
5. "Okumayı ve işlemeyi durdur", "okumayı ve işlemeyi durdur ve hafızayı başa al", ve "altprogram çağır" gibi denetleyici tarafından yapılması gereken işlemleri tanımlayan deyimler.

**NC PROGRAM**

```

00001;
N1G20G90G92X0Y0Z1.5;
N2G00X-1.375Y-1.5M03S20;
N3Z0.05;
N4G01Z-0.5F4.0;
N5Y1.375;
N6X1.375;
N7Y-1.375;
N8X-1.375
N9G00Z1.5;
N10X0Y0M05;
N11M30;

```

İŞLEME PLANI No: 0001

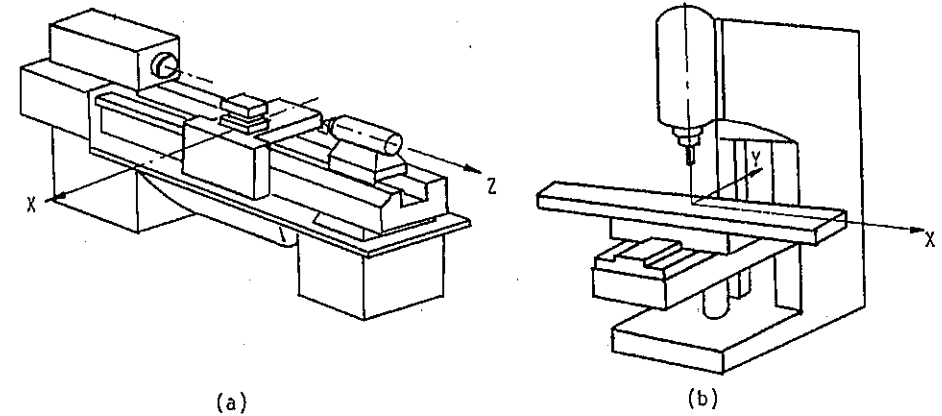
- Adım 1 : Takımı parçanın merkezine yüzeyden 1.5" yukarıya yerleştir. Mutlak koordinatları, inç olarak bir takım konumunu belirlemek için kullan. (N1 deyimini)
- Adım 2 : Takımı yatay olarak parçanın sol alt köşesine hareket ettir, (x=-1.375, y=-1.5) ve iş milini çabıştır. İş mili hızını maksimum hızın %20'sine ayarla. (N2 deyimini)
- Adım 3 : Takımı hızlı olarak aşağıya (z=0.05) noktasına indir ve istenilen derinliğe 4 in/dak. hızla in. (N3 ve N4 deyimleri)
- Adım 4 : Parçanın çevresini işle (NC'ten N8'e kadar olan deyimler)
- Adım 5 : Takımı hızlı olarak yukarı kaldır. (N9 deyimini) Başlangıç konumuna dön ve iş milini durdur. (N10 deyimini)
- Adım 6 : Bu işleme planını sonlandır. (N11 deyimini)

Şekil 2-1 Sözcük adres bloğu formatında açıklanmalı basit bir NC programı

Her deyim ve komut bir satır tutar ve genellikle *blok* olarak adlandırılır. Her deyim sonunda blok sonu kodu sembolü yer almalıdır, bu FANUC 6MB kontrol birimi için ";" ve FANUC 6T kontrol birimi için "*" olarak kullanılmaktadır. NC teknolojisinin geliştirilmesinde ileride açıklanacak 4 ayrı blok formatı kullanılmıştır.

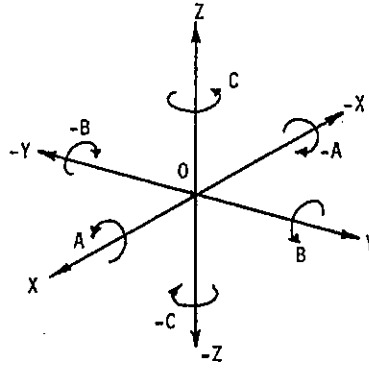
2.1 NC TEZGAHLARIN KOORDİNAT SİSTEMLERİ

Bir takım tezgahı gerçekleştirebileceği hareketlerle karakterize edilir. Takımın ve iş parçasının birbirlerine göre konumlarının değişimi gibi bu tür hareketler doğrusal yer değiştirmeler ve değişik eksenler etrafında dönme hareketlerinden oluşur. Bununla beraber, kesme hareketini sağlamak için kesicinin veya iş parçasının dönmesini sadece kapsamaz. Örnek olarak bir torna iş mili eksenine dik ve paralel sadece iki doğrusal harekete sahiptir (Şekil 2-2[a]'da görülen X ve Z eksenleri) bunun yanında bir dik freze tezgahı uç hareket eksenine sahiptir (Şekil 2-2[b]). İş milinin dönmesi (örnek olarak torna için iş parçasının freze için kesicinin dönmesi) genellikle karakteristik bir hareket olarak sayılmaz.



Şekil 2-2 Hareket eksenleri. (a) Torna. (b) Düşey freze tezgahı.

Sağ el koordinat sistemi, takımın veya iş parçasının konumunu ve hareketini tanımlamak için kullanılır. Hareket eksenleri ABD'de de hem Elektronik Endüstri Kurumu (EIA) hem de Uluslararası Standardlaşma Organizasyonu (ISO) tarafından standardlaştırılmıştır.^{21,22} Genel olarak kesme kuvvetini uygulayan iş mili eksenini, Z hareket eksenini olarak bilinir ve iş parçası ile takım tutucusu arasındaki boşluğu arttıran hareketin yönü artı yön olarak tanımlanır. X eksenini, Z eksenine diktir ve



Şekil 2-3 NC makinelerde kesici hareketini tanımlamak için sağ koordinat sisteminin kullanılması

takımın veya iş parçasının konumlandığı düzlemdeki ana hareket eksenidir. Z ve X eksenleri saptandığında, Y eksenini sağ-el kuralını temel alarak yerleştirilebilir. NC tezgahların bazıları X, Y ve Z eksenleri boyunca kesme hareketini korumak için olanlardan başka hareketlere sahip olabilir, bunlar karşılıklı olarak A, B ve C eksenlerindeki hareketler olarak tanımlanır (Şekil 2-3). Örnek olarak, dik freze tezgahının döner tablasının dönüşü eğer döner tabla eksenini iş mili eksenine paralel ise C ekseninde hareket olarak tanımlanır. Burada listelenenlerin dışındaki ek doğrusal ve dönme hareketleri; işleme merkezleri, delik işleme tezgahları, vargel işleme üniteleri ve nicelerin de sağlanmış olabilir, karşılıklı gelen eksenler 21 ve 22 numaralı referanslarda verilmiştir.

2.2 TEZGAH KODU

Şekil 2-1'den görüleceği gibi, bir program tamamlanacak kesici hareketlerini ve işlemleri tanımlayan sıralı listelenmiş deyim (veya blok) serilerinden oluşmaktadır. Bir deyim kelimelere de bölünebilir. Karakterler, sınırlayıcılar ve sayısal basamaklar bir kelimeyi oluşturan elemanlardır. Örnek olarak, Şekil 2-1'deki programdaki şu deyim;

N1 G20 G90 G92 X0 Y0 Z1.5;

"N1," "G20," "G90," "Gaz" "X0," "Y0," "Z1.5"; kelimelerinden oluşmaktadır ve bu deyimde kelimelerin elemanları "N", "G", "X", "Y", "Z", "1", "2", "5", "9", "0", "." ve ";" dir.

İyi bilindiği üzere, telgraf mesajı Morse Kodunu kullanan bir insan tarafından iletilmekte ve anlaşılmalıdır. Benzer olarak, bir NC programı da ikili kodlu ondalıklı kodun kullanımı ile anlaşılabilir. Halihazırda NC tezgahlarda iki adet

kullanılan kod vardır; EIA RS-244-B kodu²³ ve ASCII kodu (Bilginin Değişimi için Amerikan Standard Kodu). Bunlar Şekil 2-4'de gösterilmiştir. NC programının NC denetleyicisine aktarıldığı geleneksel ortam kağıt şerittir. Bilgisayarların üretimde giderek artan kullanımı ile, bir NC programı bir bilgisayardan da belleğe veya bir NC program hazırlama sisteminden doğrudan CNC denetleyicisine RS232 kablo ile yüklenebilir. ASCII kodu RS358-B kodu²⁴ gibi, hem EIA hem ISO tarafından da kabul edilmiştir. Her iki kodda Bir çok CNC tezgah tarafından kabul edilmektedir. Bunlar EIA kodu (RS-244-B kodudur) ve ISO kodudur (ASCII kodundan doğmuştur).

Bu iki kodun da detaylı tanımlamaları bir çok kitapta bulunabilir (örneğin 23,24 ve 25 numaralı referanslar). Bu iki kod arasındaki fark, EIA kodunda şerit üzerine delinen deliklerin sayısının tek sayı olması, ISO kodunda ise çift sayı olmasıdır. Bu nedenle karşılıklı olarak tek ve çift eşliklidirler. EIA RS-244-B kodundaki Kanal 5, ISO kodundaki Kanal 8 eşlik kullanımı için ayrılmıştır. Bu kanallara delik sadece, bir karakteri veya basamağı temsil eden deliklerin sayısı gereken tek veya çift eşliği sağlayamazsa delinir. Yukarıda anlatılan farktan başka, bu iki kodda aynı karakteri veya basamağı belirtmek için farklı delik kombinasyonları kullanılır. Örnek bir delikli şerit Şekil 2-5'te sunulmuştur.

2.3 BİR NC PROGRAMININ BLOK FORMATLARI

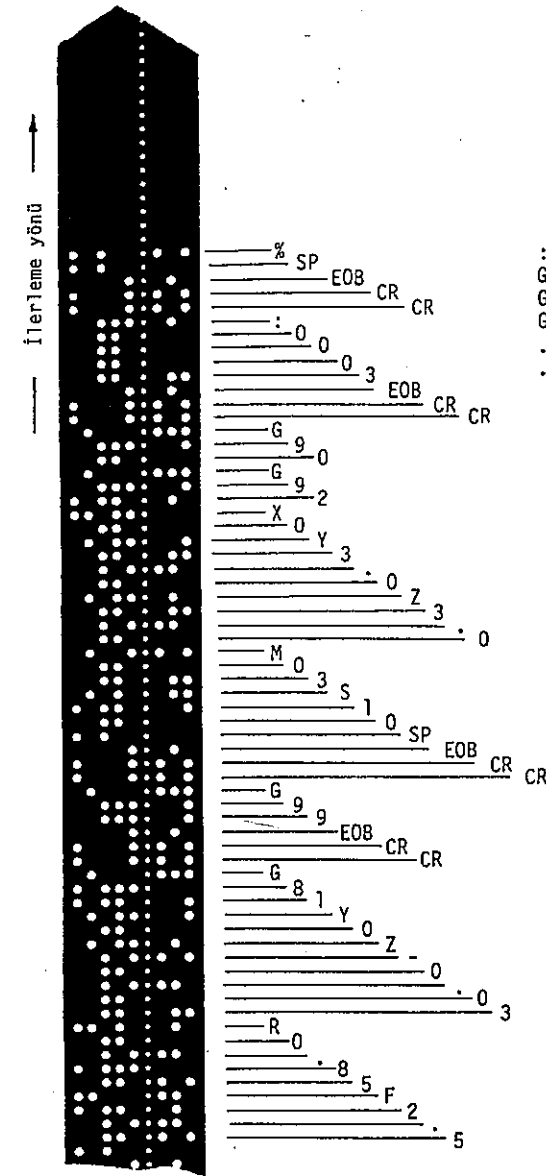
NC sözcüklerini bir deyim (veya blok) içerisinde kullanmanın NC denetleyicisinin tasarımına bağlı olarak bir çok farklı yolu vardır. İlk NC formatlarından biri "sabit sıralı" olandır. Burada bir bloktaki her kelime için belirli bir sıra gereklidir ve denetleyici tarafından kullanılan her kelimenin NC tezgahın durumunda anlamlı bir değişikliğe neden olsalar veya olmasalar da bir blok içinde olması gereklidir. Her kelime NC'li tezgahın durumunu belirten bir numaradan oluşur (X yönündeki konum, ilerleme hızı veya devir sayısı gibi). Bundan dolayı bir deyim sabit sıralı listelenmiş ve farklı NC kelimeleri temsil eden sabit sayıdaki numaralardan oluşur. Örnek olarak, eğer bir NC kelimesini belirtmenin sırası şu şekilde ise;

deyim no — hareket türü — X koordinatı — Y koordinatı — Z koordinatı — ilerleme hızı — devir sayısı — diğer fonksiyonlar (örneğin iş mili açık/kapalı) sabit sıralı formatlı bir deyim şu şekilde gözükabilir;

003 01 1.00 2.00 3.00 2.0 500 3

003 etiketlenmiş deyim, takımın mevcut konumundan (1.0, 2.0, 3.0) konumuna 2.0 in/dak hızla doğrusal olarak hareket iş mili saat yönünde 500 dv/dak hızla dönerek

ISO code								EIA code								Mesning		
Charter	8	7	6	5	4	3	2	1	Charter	8	7	6	5	4	3		2	1
0									0									Sayıt 0
1									1									Sayıt 1
2									2									Sayıt 2
3									3									Sayıt 3
4									4									Sayıt 4
5									5									Sayıt 5
6									6									Sayıt 6
7									7									Sayıt 7
8									8									Sayıt 8
9									9									Sayıt 9
A									a									Adres A
B									b									Adres B
C									c									Adres C
D									d									Adres D
E									e									Adres E
F									f									Adres F
G									g									Adres G
H									h									Adres H
I									i									Adres I
J									j									Adres J
K									k									Adres K
L									l									Adres L
M									m									Adres M
N									n									Adres N
O									o									Adres O
P									p									Adres P
Q									q									Adres Q
R									r									Adres R
S									s									Adres S
T									t									Adres T
U									u									Adres U
V									v									Adres V
W									w									Adres W
X									x									Adres X
Y									y									Adres Y
Z									z									Adres Z
DEL									Del									iptal (deği hatasının iptali)
NUL									Blank									Değinnemiş. ETA kodunun özel bölümünde kullanılmaz
BS									BS									Arka alan
HT									Tab									Tabulatör
LF or NL									CR or EOB									Blok sonu
CR																		Taşıyıcının geri gelmesi
SP									SP									Bosluk
%									ER									Mutlak geri sarmayı durdurma
((2-4-5)									Kontrol çıkışı (bir açıklama sonu)
)									(2-4-7)									Kontrol girişi (açıklama sonu)
+									+									Pozitif işaret
-									-									Negatif işaret
:									:									Kolon
/									/									Seçimli blok atlama
.									.									Periyod (ondañık bir nokta)
#									#									Şekil
\$									\$									Önler işareti
&									&									Amper sand (ve)
*									*									Apostrof
.									.									Yıldız
:									:									Virgül
;									;									Noktalı virgül
<									<									Açılı sol parantez
=									=									Eşit
>									>									Açılı sağ parantez
?									?									Soru işareti
@									@									Reklam işareti
^									^									Tırnak
																		Sol baş işareti
~									~									Sağ baş işareti

Şekil 2-4 NC Delikli şerit için EIA ve ISO kodları.²⁶ (FANUC Ltd.)

Şekil 2-5 Bir NC delikli şeridi

gitmesi gerektiği anlamına gelir. Böylece tam bir NC programı numaraların sıralarından oluşur. Bu da sayısal denetim teriminin nereden doğduğunu göstermektedir.

NC teknolojisinin geliştirilmesi sırasında kullanılan ikinci tür blok formatı da blok adres formatıdır, ki burada tezgaha hangi kodların değişeceğini, belirtileceğini

veya her ikisini bu komutta bildirmek için bir blok kodu, bloğun başında belirtilir. Tablo 2-1 farklı çalışma gereksinimleri için blok kodlarının bir örneğini vermektedir. Bu tabloya göre bir NC deyimindeki blok adres kodu olan 20 bu blokta sadece X ve Y yönlerinde bir değişiklik olacağı anlamına gelir. Sabit sıralı blok formatında olduğu gibi bir bloktaki bütün sözcükler sabit sırada belirtilmiştir. Değişmeyecek sözcüklerin deyimde yer almasına gerek olmadığından blok adres formatı sabit sıralı olandan daha derli topludur.

Tablo 2-1 BLOK ADRES KODLARINA ÖRNEK

Değişiklik	Konumda veya yönde değişiklik				
	X	Y	X,Y	Z	aynı
Konum	00	10	20	30	
İlerleme hızı	01	11	21	31	41
Hız (Devir sayısı)	02	12	22	32	42
Soğutucu	03	13	23	33	43
İlerleme hızı, hız	04	14	24	34	44
İlerleme hızı, soğutucu	05	15	25	35	45
Hız, soğutucu	06	16	26	36	46
İlerleme hızı, hız, soğutucu	07	17	27	37	47

Örnek : Eğer bir programda ilerleme hızı ile, X ve Y yönündeki konum değiştirilecekse blok adres kodu 21'dir.

Kaynak: C.R Asfahl. Robots and Manufacturing Automation. Copyright © 1985 by John Wiley F Sons, Inc., ve reprinted by permission of John Wiley F Sons, Inc.

Üçüncü format sabit sıralı formatın bir değişimi olan etiket sıralıdır. Bu formatta, her sözcükten önce mutlaka bir etiket karakteri belirtilmelidir. Aynı şekilde sözcükler de belirli bir sırada olmalıdır. Eğer bir önceki duruma göre belli bir sözcükte hiç bir değişim yoksa sözcüğün içeriği tamamen çıkarılabilir. Bununla beraber bütün sözcüğün çıkarılmasına rağmen belli bir sözcükten önceki etiket koduna hala ihtiyaç vardır.

Yukarıda belirtilen üç formatda eski kablolu nümerik kontrol sistemlerinde kullanılmıştır. Bilgisayarlı nümerik kontrol sistemlerinin gelişmesi ile sözcük adres formatı tarafından bir kenara itildiler ve işe yaramaz olarak kabul edilebilirler.

Çağdaş CNC kontrol sistemlerinde kullanılan ise sözcük adres formatıdır. Her deyimdeki (bloktaki) kelime, kelimenin adresini veya anlamını belirten karakterlerden ve kapsamını belirten numaralardan oluşur. Örnek olarak X=1.0 takım konumunu belirleyen kelime şu şekildedir;

X1.0

FANUC 6MB ve 6T kontrol sistemlerindeki adresler ve anlamları karşılıklı olarak Tablo 2-2 ve 2-3'te listelenmiştir.^{26,27} Bu iki tablodan görüleceği gibi karakterden sonra gelen numara kelimenin veya kodun türüne göre tam veya reel sayı olabilir. X,Y,Z,I,J,K,R,Q,U,ve W kodları koordinat konumlarını belirler ve ondalık noktası olmayan sayıları da kabul edebilirler. Bu durumda sayının büyüklüğü belirtilen sayının denetleyicinin çözünürlüğü (0.001 mm metrik programlama, inç programlama için 0.0001 inç) ile çarpılmış gibi kabul edilir. Örnek olarak eğer programlama inç birimi ile yapılmışsa;

X1. X=1.0 inç
X1 X=0.0001 inç
X100 X=0.0100 inç anlamına gelir.

Tablo 2-2 FANUC 6MB NC KONTROL SİSTEMLERİ İÇİN SEÇİLMİŞ KELİME ADRES KODLARI VE FORMATLARI

Adres Kodu	Anlamı	Adres kodunu izleyen numaranın türü	Rakamların basamakları
;(ISO) / O (EIA)	Program numarası	Tamsayı / 4 basamaklı	
N	Deyim etiketi	Tamsayı / 1'den 4'e kadar basamaklı	
G	Hareketi tanımlayan kod	Tamsayı/iki basamaklı	
X,Y,Z*	X,Y ve Z yönlerinde koordinatlar	İnç giriş; reel sayı ondalık noktasının sağında ve solunda 4 basamaklı	
R*	Yayın veya dairenin yarıçapı	Metrik giriş; ondalık noktasının önünde 5 arkasında 3 basamaklı	
I,J,K*	Yay merkezinin koordinatları	İnç giriş için nx0.0001 inç metrik giriş için nx0.001mm sayılır.	
Q,R*	Çevrim deyimindeki konum parametreleri		
F*	İlerleme hızı veya vida adımı	İnç/dak giriş için; reel sayı, noktadan evvel 3 basamaklı noktadan sonra 2 basamaklı mm/dak giriş; tamsayı, 5 basamaklı İnç/dev giriş; reel sayı, ondalıktan önce 2 basamak, sonra 4 basamak mm/dev giriş; reel sayı, ondalıktan önce 3, ondalıktan sonra 2 basamaklı.	
S	İş mili hızı (Devir sayısı)	Tamsayı, NC'li tezgahın yapısına bağlı olarak 2 veya 4 basamaklı	
M	Tezgahın çalışması	Tamsayı, 2 basamaklı	
D	Ofset sayısı	Tamsayı, 2 basamaklı	

P,X*	Bekleme süresi (sn)	Reel sayı, ondalık noktadan evvel 5 basamaklı, noktadan sonra 3 basamaklı
P	Programın ismi veya deyim numarası	Tamsayı/4 basamaklı
L	Tekrarlama sayıcı	Tamsayı, 1 ile 4 basamaklı
:	Blok (deyim) sonu	

Örnek olarak, bir deyim şu şekilde olabilir; N1G92X2.OY1.354Z05M03S40

* Ondalık noktadan sonraki ve ondalık noktadan önceki sıfırlar kullanılmayabilir.

Metrik programlamada;

X1.	X=1.0 mm	
X1	X=0.001mm	
X100	X=0.100mm	anlamına gelir.

Elle NC programlamada hata yapılmasını engellemek için ondalık noktanın kullanılması önerilir.

Tablo 2-3 FANUC 6T NC KONTROL SİSTEMLERİ İÇİN SEÇİLMİŞ KELİME ADRES KODLARI VE FORMATLARI

Adres Kodu	Anlamı	Adres kodunu izleyen numaranın türü	Rakamların basamakları
;(ISO) / O (EIA)	Program numarası	Tamsayı / 4 basamak	
N	Deyim etiketi	Tamsayı / 1'den 4'e kadar basamaklı	
G	Hareketi tanımlayan kod	Tamsayı/iki basamaklı	
X,Y,Z*	X,Y ve Z yönlerinde koordinatlar	İnç giriş; reel sayı ondalık noktasının sağında ve solunda 4 basamaklı	
U, W	X ve Z eksenlerindeki ek koordinatlar	Metrik giriş; ondalık noktasının önünde 5 arkasında 3 basamaklı	
R*	Yayın veya dairenin yarıçapı	İnç giriş için nx0.0001 inç metrik giriş için nx0.001mm sayılır.	
I,K*	Yay merkezinin koordinatları veya radyal veya eksenele geri çekilme		
D*	Talaş derinliği		
F*	İlerleme hızı veya vida adımı	İnç/dev giriş; reel sayı, ondalıktan önce 2 basamak, sonra 4 basamak mm/dev giriş; reel sayı, ondalıktan önce 3, ondalıktan sonra 2 basamaklı	

E*	Vida adımı	İnç/dev giriş; reel sayı, ondalıktan önce 1 basamak, sonra 6 basamak. mm/dev giriş; reel sayı, ondalıktan önce 3, ondalıktan sonra 4 basamaklı.
S	İş mili hızı (Devir sayısı)	Tamsayı, 2 veya 4 basamaklı
T	Takım ve telafi numarası	Tamsayı, 4 basamaklı, ilk 2 sayı takım numarası, son iki sayı ofset sayısı
M	Tezgahın çalışması	Tamsayı, 2 basamaklı
X,U ve P*	Bekleme süresi (sn)	Reel sayı, ondalık noktadan evvel 5 basamaklı, noktadan sonra 3 basamaklı. Şu girişlerin hepsi eşittir; 3.0-3000-3 seconds
P, Q	Programın ismi veya deyim numarası	Tamsayı, 4 basamaklı
A	Diş açısı	Tamsayı (kademeli sayı)
L	Tekrarlama sayıcı	Tamsayı, 1 ile 4 basamaklı
L	Tekrarlama sayıcı	Tamsayı, 1 ile 4 basamaklı
:	Blok (deyim) sonu	

* Ondalık noktadan sonraki ve ondalık noktadan önceki sıfırlar kullanılmayabilir.

F,E ve P kodları aynı şekilde reel sayı girişine de izin verirler. Bunlar gelecek bölümlerde tartışılacaktır. Bütün diğer kodlar tamsayı girişini kabul eder.

Kelime adres formatının önemli unsurlarından biri de kelimenin anlamının ilk alfabetik karakter tarafından net olarak belirtildiğinden dolayı bloğun içindeki kelimelerin sabit dizili olarak belirlenmesi zorunluluğunun olmamasıdır. Ek olarak, eğer bir kod bir önceki deyimdeki ile aynı ise normalde tekrar yazılmayabilir.

2.4 TORNALAMA VE FREZELEME İŞLEMLERİ İÇİN NC PROGRAMLAMADA KULLANILAN TEMEL KODLAR

Bir NC programı genellikle iki ayrı komut türünden oluşur. Birinci tür, kesici hareketlerini iş parçasına göre tanımlamak için kullanılır. Kesici hareketleri aşağıdaki parametrelerle tanımlanır:

1. NC programında kullanılan koordinat sisteminin türü (mutlak veya artışı), birimi (metrik veya inç) ve bağıl konumları.
2. Kesici hareketlerinin türleri: konumlama, doğrusal interpolasyon, dairesel interpolasyon, vida açma veya diğerleri.
3. İlerleme hızı, bekleme süresi ve özel bir takımın iş parçasını kesmesi için verilmesi gereken telafi miktarı.

Tablo 2-2'den görüleceği gibi, FANUC 6MB kontrol birimine sahip 3 eksenli bir freze tezgahında kesici hareketlerini tanımlamak için aşağıdaki kodların biri veya çoğuna gerek olduğu açıkça bellidir: G, X, Y, Z, R, I, J, K, H, D, P, Q, R ve F.

İkinci tür komutlar ise NC'li tezgahın çalışmasını kontrol etmek, takım veya iş mili hızını seçmek program ve deyim numaralarını tanımlamak için kullanılır. FANUC 6MB kontrol birimi üzerinde bu amaçlarla şu kodlar kullanılır: M, T, L, O, P ve S.

FANUC 6T kontrol sistemli bir torna ile kullanılan kodlar (Tablo 2-3) bir freze tezgahındakilerden çok az farklıdır. Bir torna sadece iki boyutta (X ve Z) harekete sahiptir. Buna göre de kesicinin konumunu belirleyen kodlar sadece bu iki yönde gereklidir. FANUC 6T kontrol sistemi X ve Z eksenlerindeki artışlı koordinatları tanımlamak için karşılıklı olarak U ve W olan iki ayrı kod kullanır. Ek olarak bir torna kesici takımı için gereken kayma T kodu tarafından belirlenir. Bundan dolayı FANUC 6T kontrol sisteminde kesici hareketlerini tanımlayan kodlar G, X, Z, U, W, R, I, K, P, Q, D ve F'dir. Tornanın çalışmasını tanımlayan kodlar bir freze tezgahının kodları ile takımı ve telafiyi tanımlayan T kodu hariç temel olarak aynıdır.

G ve M kodlarından başka X, U ve P gibi birden çok fonksiyona sahip olanlar hariç diğer bütün kodların tek bir fonksiyonu vardır. Tablo 2-4, 2-5 ve 2-6'dan görüleceği gibi G ve M kodları bir çok fonksiyona sahiptir.

Tablo 2-4 FANUC 6MB KONTROL SİSTEMİNDE KULLANILAN SEÇİLMİŞ G KODLARI:

G kodu	Grup No.	Fonksiyon
G00	01	Hızlı konumlandırma
G01		Doğrusal enterpolasyon hareketi
G02		Dairesel enterpolasyon hareketi (saatin dönme yönünde)
G03		Dairesel enterpolasyon hareketi (saatin ters yönünde)
G04	00	Bekleme süresi
G10		(offset) Telafi değerini ayarlama
G17*	02	X-Y düzleminin seçimi
G18		Z-X düzleminin seçimi
G19		Y-Z düzleminin seçimi
G20	06	İnç giriş
G21		Metrik giriş
G28	00	Referans noktasına dönüş
G40*	07	Kesici telafisinin iptali
G41		Soldan kesici-yarıçapı telafisi
G42		Sağdan kesici-yarıçapı telafisi

G80*	09	G80 kodları tarafından belirlenen çevrim hareketinin iptali
G81		Delik Delme
G83		Kademeli delme çevrimi
G85		Delik işleme çevrimi
G90	03	Mutlak koordinatlarla programlama
G91*		Artışlı koordinatlarla programlama
G92	00	Tezgaah koordinat sisteminin orijini ayarlama
G94	05	Dakikadaki ilerleme
G95		Bir devirdeki ilerleme
G98	10	Çevrim hareketinde ilk noktaya dönme
G99		Çevrim hareketinde R noktasına dönme

* Sistem açıldığında işaretlenmiş kodlar kullanımdadır.

Tablo 2-5 FANUC 6T KONTROL SİSTEMİNDE KULLANILAN SEÇİLMİŞ G KODLARI:

G kodu	Grup No.	Fonksiyon
G00	01	Hızlı konumlandırma
G01		Doğrusal enterpolasyon hareketi
G02		Dairesel enterpolasyon hareketi (saatin dönme yönünde)
G03		Dairesel enterpolasyon hareketi (saatin ters yönünde)
G04	00	Bekleme süresi
G10		Telafi değerini ayarlama
G20	06	İnç giriş
G21		Metrik giriş
G28	00	Referans noktasına dönüş
G32	01	Diş açma çevrimi
G40*	07	Kesici uç yarıçapı telafisinin iptali
G41		Soldan kesici uç yarıçapı telafisinin
G42		Sağdan kesici uç yarıçapı telafisi
G50	00	Tezgaah koordinat sisteminin orijini ayarlama
G70		(Son talaş) ince işleme çevrimi
G71		Tornalamada talaş kaldırma
G72		Alın işlemede talaş kaldırma
G73		Tekrarlayan dizin
G76		Diş açma çevrimi (çok çevrimli)
G98	05	Dakikadaki ilerleme
G99		Bir devirdeki ilerleme

* Sistem açıldığında işaretlenmiş kodlar kullanımdadır.

Tablo 2-6 FANUC KONTROL SİSTEMİNDE KULLANILAN SEÇİLMİŞ M KODLARI:

M kodu*	Fonksiyon
M00	Programı durdurma (genellikle programın ortasında kullanılır.)
M01	İsteğe bağlı durdurma (Bu fonksiyon sadece kontrol paneli üzerindeki isteğe bağlı durdurma düğmesi basılı ise etkilidir.)
M02	Programın sonu
M03	İş milinin saatin yönünde çalışmaya başlaması (freze tezgahı) İş milinin ileriye dönmesi (torna tezgahı)
M04	İş milinin saatin tersi yönde çalışmaya başlaması (freze tezgahı)
M05	İş milinin geriye dönmesi (torna)
M08	İş milini durdurma
M09	Soğutma sıvısını açma
M11	Soğutma sıvısını kapatma
M30	Programın sonu veya belleği (veya şeridi) geri sarma
M41	İş mili hızının düşük devir aralığı
M42	İş mili hızının yüksek devir aralığı
M98	Bir alt programı çağırma
M99	Ana programa dönüş

* Farklı üreticilerin farklı tezgahlarında M kodları farklı olabilir. Örneğin, M06 kodu normal olarak diğer makinelerde takım değişimini belirtir. Okuyucu kendi özel tezgahının programlama kitabına başvurmalıdır.

Şunu belirtmeliyiz ki, sadece içinde tanımlandığı blokta mı etkili olduğuna yoksa ileride açıkça belirtilerek iptal edilene kadar mı geçerli olduğuna bağlı iki farklı G kodu türü vardır. 00 grubundan başka bir grubun içinde yer alan kod bir kere belirlendiğinde, aynı gruptaki başka bir kod tarafından iptal edilene kadar geçerlidir. Bunlara "modal" denilir. 00 grubunda yer alan herhangi bir kod modal değildir. G kodunun her grubunun bir hareket türünü veya bir çalışma konumunu temsil ettiği görülebilir. Kontrol birimi açıldığında asteriksle işaretlenmiş G kodları etkilidir. Aynı gruba dahil olmayan G kodları bir blok içerisinde belirtilebilir. Eğer bir blok içerisinde aynı gruba bağlı birden fazla G kodu belirlenirse sadece son belirlenmiş olan etkilidir.

Örneğin Şekil 2-1'de görülen NC programının ilk deyiminde farklı gruplara ait (06,03 ve 00) üç G kodundan oluşmaktadır. Aşağıdaki deyim;

G02G00X_Y_J_;

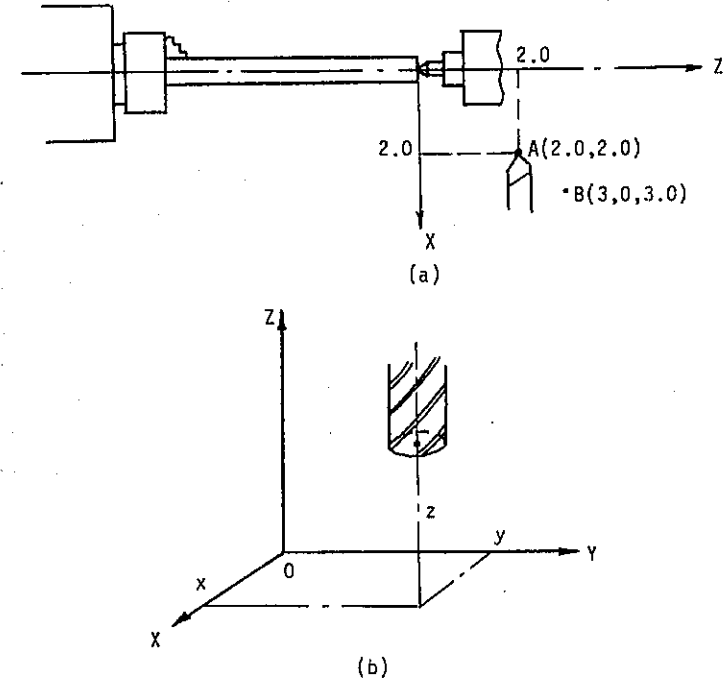
1. gruptan iki G koduna sahiptir. Bundan dolayı hareketi en son tanımlanmış kod olan G00 belirler. Buna göre I ve J kodları iptal edilir ve bu nedenle hareket

G00X_Y_;
tarafından tanımlanır.

Şunu belirtmeliyiz ki, bir kural olarak aynı grupta yer alan G kodlarının birden fazlasının bir arada belirtilmesinden kaçınılmalıdır. Ek olarak, bu kurala ait istisnalar da vardır; farklı gruplarda yer alan G kodları aynı blok içerisinde birarada kullanılabilirler. Örneğin aşağıdaki deyimde;

G04G00X_Y_;

farklı gruplardan iki G kodu vardır. G04 kodu takma bulunduğu konumda beklemesini, G00 kodu da bulunduğu konumdan X ve Y kodları ile belirlenen konuma hareket etmesini emreder. Bu iki kodun etkisi birbirleri ile çelişmektedir. Grup 00 ve 01'de yer alan G kodlarının birarada kullanılmaması önerilir. G kodlarının dışındaki kodlar bir blok içerisinde sadece bir kere belirlenebilirler. Kelime adres formatı ile bir blok içindeki kodlar herhangi bir sırada belirlenebilir.



Şekil 2-6 Bir torna için koordinat sistemi ve X ve Z kodları. Koordinat sisteminin başlangıcı, takımın A konumuna dayalı G50x4.0Z 2.0 deyiminiyle ayarlanabilir. (b) Bir freze tezgahı için koordinat sisteminin başlangıcı G92X.rYyZz deyiminiyle ayarlanır.

Tornada kullanılan X kodunun frezede kullanılan farklı anlamı olduğuna dikkat edilmelidir. Genel olarak, tornada işlenecek parçanın kesitleri silindriktir. Çizimde silindirik bir yüzeyin yarıçapı genellikle yarıçap ile değil çapı ile tanımlanır ki bu sayede, kolayca ölçülebilir. Bu yüzden X kodu çap değeri olarak tanımlanır (X koordinat değerinin iki katı) böylelikle kesici konumları mühendislik çizimlerinden daha kolay bir şekilde belirlenebilir. Şekil 2-6(a)'daki A(2.0,2.0) olan takım konumunu belirten kodlar şu şekildedir;

X4.0Z2.0

X koordinatının artışlı biçimi olan U kodu da çap değeri olarak belirtilmelidir. Böylece takımı A noktasından B noktasına hareket ettirmek için kullanılacak U ve W kodları şunlardır;

U2.0W2.0

2.5 KOORDİNAT SİSTEMİNİN TANIMLANMASI

Bir hareket deyimi belirtilmeden önce, bir koordinat sistemi tanımlanmalıdır. Bir NC'li tezgahta koordinat ekseninin yönleri değişmezdir. Bununla beraber koordinat sisteminin orijini istenilen herhangi bir konuma ayarlanabilir. Bir kesicinin hareketi sadece koordinat sisteminin orijini doğru konuma ayarlandıktan sonra doğru olarak tanımlanabilir. G92 kodu aşağıda belirtilen formatta kullanılarak koordinat sisteminin orijini tanımlanabilir; (Şekil 2-6(b))

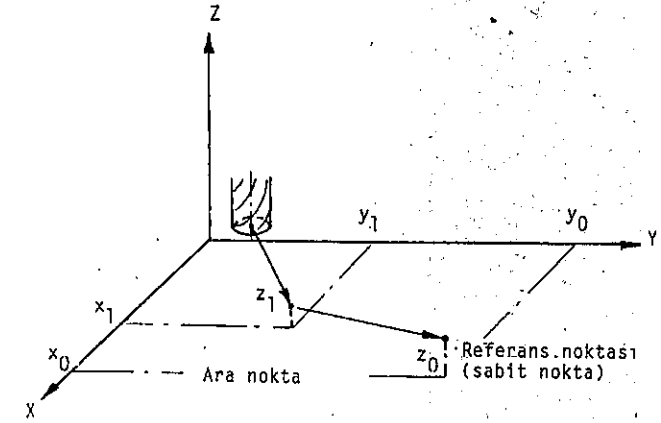
G92XxYyZz

x, y, ve z değerleri tanımlanacak koordinat sistemindeki takımın bulunduğu yerin koordinatlarıdır. Şekil 2-1'de verilen örnekte koordinat sisteminde takımın ilk konumu olan (0.0,1.5) tanımlanacaksa, koordinat sistemini tanımlayan deyim şudur;

G92X0Y0Z1.5;

FANUC 6T kontrol sistemli bir tornada koordinat sisteminin orijinini ayarlamak için G92 yerine G50 kodu kullanılır. Bir örnek Şekil 2-6(a)'da sunulmuştur.

Koordinat eksenini tanımlamanın başka bir yolu da orijini NC tezgahının sabit referans noktasına göre ayarlamaktır (Şekil 2-7). Bir NC programında takım ilk referans noktasına hareket eder ve sonra G50 (torna için) veya G92 (freze için) kodu



Şekil-7 G28XxYyZz ve G92XxYyZz deyimleri kullanarak referans noktasına bağlı olarak koordinat sisteminin ayarlanması.

kullanılır. Bu sayede takımın başlangıç konumu bir işlemten diğerine değişse bile koordinat sisteminin orijini her zaman aynı noktada tanımlanabilir.

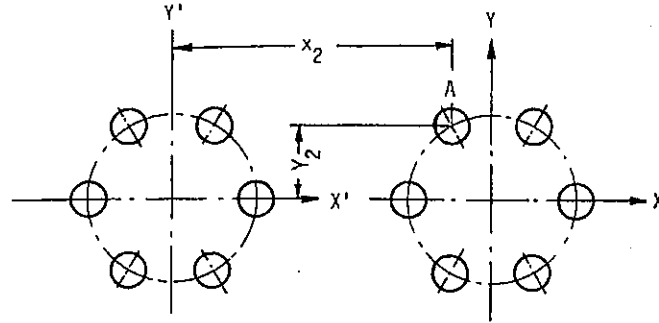
Bu metod özellikle NC'li tezgah üzerinde birbirinin aynı parçaların işlenmesinde çok kullanışlıdır. Takım, program çalıştırılmadan önce istenen herhangi bir konuma hareket ettirilebilir. Takımı referans noktasına getirmek için FANUC CNC kontrol sistemlerinde özel G28 kodu sağlanmıştır, ve formatı şu şekildedir;

G28XxYyZz (freze tezgahı için)
G28 { XxZz } (torna tezgahı için)
 { UuWw }

ki burada x, y ve z ara noktanın koordinatlarıdır. Bu kod tarafından tanımlanan takım hareketi, bulunan konumdan, ara konuma, oradan da referans konumuna hızlı bir harekettir. Artışlı programlama da x, y, z, u ve w bulunan takım konumundan ara noktaya olan artışlı koordinatlar olmalıdır. Eğer ara noktaya gerek yoksa bu koordinatlar sıfır olmalıdır. Şekil 2-7'deki gereken koordinat sistemi aşağıdaki iki deyimle ayarlanabilir;

G90G28Xx1Yy1Zz1;
G92Xx0Yy0Zz0;

NC'li tezgahın referans noktası genellikle tezgah tablasının bir köşesine ayarlanır. Bu aynı zamanda takımın değiştirmek için güvenli bir geri çekme konumudur. Belirlenen ara nokta ile, başlangıç ve referans noktaları arasındaki engellere (parça veya bağlama aparatı gibi) çarpmanın önlenmesi için kesici yolu tanımlanır.



Şekil 2-8 Kesici yolunun tanımı, değişik koordinat sistemleri kullanılarak basitleştirilebilir.

Kesici yolunu tanımlamayı kolaylaştırmak için NC programında kullanılmış koordinat sistemi değiştirilebilir. Şekil 2-8'de iki delik setinin iki farklı koordinat sisteminde, isimleri ile X, Y, Z ve X', Y', Z', daha kolay tanımlanabildiğini gösteren bir örnektir. Bundan dolayı bir NC programında tezgah koordinat sistemini X, Y, Z olarak tanımlayabiliriz. İlk delik setini delme işlemi için gerekli kesici hareketlerini tanımladıktan sonra aşağıdaki deyim ile;

G92X₂Y₂Z₂;

tezgahın koordinat sistemini X' - Y' - Z' olarak, son takım konumu olan A noktası ile X' - Y' - Z' koordinat sisteminde x₂, y₂, z₂ koordinatları ile tanımlayabiliriz.

FANUC 6T kontrol sistemine sahip bir tornada G92 kodu yerine G50 kodunu kullanılır ve deyimde sadece X ve Z kodları dahil edilir. Torna da kesici yollarını programlamak için genellikle işlenen silindirik parçanın alın yüzeyi referans yüzeyi olarak kabul edilir. Böylece alın yüzeyi işlendikten sonra merkezi parçanın tezgah koordinat sisteminin orijini olarak kabul edilebilir.

Şu belirtilmelidir ki, tezgah koordinat sisteminin takımın bulunduğu yer referans alınarak yeniden tanımlanması takım konumlama hatalarının birikmesi sonucuna neden olur. Sebebi takım tezgahının konumlama hatasından dolayı gerçek takım konumunun hiçbir zaman programlanamamasıdır. Bu yüzden bu metod sadece biriken hataların belirlenen toleransı aşmadığı durumlarda kullanılabilir.

Kesicinin konumu mutlak koordinatlarda veya artışı koordinatlarda tanımlanabilir. FANUC 6MB kontrol sistemli bir freze tezgahında bu iki deyim karşılıklı olarak G90 ve G91 kodları ile tanımlanır. Böylece aşağıdaki deyim;

G90G00X3.0Y4.0;

belirlenen bu bloktaki ve bundan sonraki koordinatların mutlak koordinatlarda belirlendiği anlamına gelir. Bunun yanında aşağıdaki deyim;

G91;

bu blokta ve bundan sonrakilerde tanımlanan koordinatların artışı olduğu anlamına gelir.

Torna için X ve Z yönlerindeki artışı koordinatları temsil etmek için U ve W kodları kullanılır. Buna göre G90 ve G91 kodları frezede oldukları gibi aynı anlamlara sahip değildirler. Artışı programlamanın NC programını kısalttığı ve basitleştirdiği bir çok durum vardır. Bu bölümün sonundaki Kısım 2.9'da yer alan Örnek 2 tipik bir durumu temsil etmektedir.

2.6 KESİCİ HAREKETLERİNİN TANIMLAMASI

2.6.1 NC'li Tezgahın Temel Hareketi

NC'li tezgahların bir çoğu üç çeşit temel hareketi yapabilir; hızlı ilerleme hareketi, doğrusal enterpolasyon hareketi ve dairesel enterpolasyon hareketi.

Hızlı İlerleme Hareketi: Bu hareket G00 kodu tarafından aşağıdaki formatla tanımlanır;

G00X_xY_yZ_z;

ki burada x, y ve z koordinatları kesicinin/işin varış noktasıdır. Bazı NC'li tezgahlar takım eşzamanlı olarak sadece iki eksenle hareket ettirebilir. Bu durumda G00 kodu ile sadece iki koordinat belirlenebilir. Eğer X, Y ve Z kodlarından biri belirtilmemişse mutlak programlama da bir önceki deyimdeki ile aynıdır. Artışı programlamada yazılmamış kod, artışı koordinatın sıfır olduğu anlamına gelir.

Hızlı ilerleme hareketinin amacı takımın bulunduğu konumdan istenen varış noktasına (x, y, z) kesici yolu üzerinde herhangi bir kısıtlama olmadan hareket ettirmek içindir. Sadece son konum işleme operasyonunda öneme sahiptir. Bu kodun tipik bir uygulaması matkabı bir konumdan öteki delme noktasına hareket ettirmektir. G00 kodu ile, takım NC kontrol birimi tarafından bir parametre ile ayarlanmış bulunan hızın aynısı ile X, Y ve Z yönlerinde bu üç koordinattan birine ulaşana kadar hareket eder. Sonra aynı hızla diğer iki koordinattan birine ulaşana kadar aynı hızla hareket eder. Daha sonra tek yönde varış noktasına ilerler. Bu

yüzden kesici yolları bulunan takım konumundan varış noktasına koordinatların artışları tarafından saptanır. Şekil 2-9 farklı varış noktaları için kesici yollarını göstermektedir. Genellikle kesici yolları artışlı koordinatların farklı eksenlerde aynı veya hareketin sadece bir yönde olduğu durumlar dışında doğrusal hatlara sahip değildir. Hızlı ilerleme esnasında hareketin hızı (ilerleme hızı) kontrol birimi tarafından ayarlandığı için ilerleme hızı olan F koduna gerek yoktur.

Artışlı programlama etken olduğunda X,Y ve Z koordinatları artışlı değerlerde verilmelidir. Böylece takım bulunduğu konumdan (1.0,2.0,3.0) (2.5,-4.2,3.0) konumuna hareket ettiren deyim şu şekildedir;

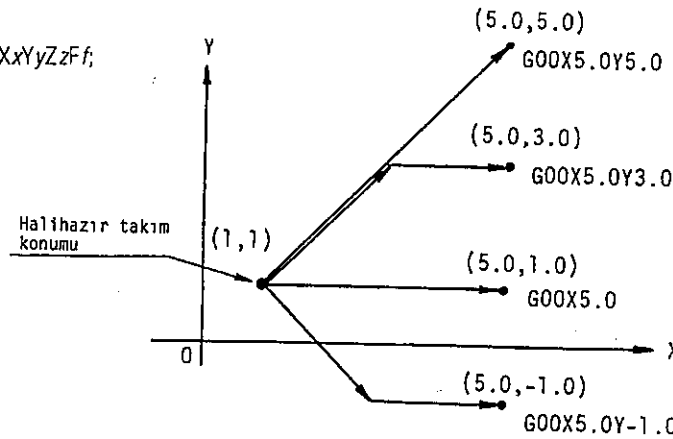
G91G00X1.5Y-6.2;

Torna için, X ve Z kodları yerine U ve W kodları kullanılmalıdır. Takımın (1.0,3.0) konumundan (2.5,-2.5) noktasına hareketi şu deyimle tanımlanır;

G00U3.0W-5.5*

Doğrusal Enterpolasyon Hareketi. Bu hareket freze tezgahı için G01 kodu ile aşağıdaki formatla tanımlanır;

G01XxYyZzFf;



Şekil 2-9 G00 kodu ile tanımlanan kesici yolu takımın, bulunduğu konumdan varış noktasına olan mesafenin X ve Y eksenlerindeki farkları ile saptanır.

Burada x,y,z varış noktasının mutlak veya artışlı koordinatları olup f ise dakikadaki inç veya dakikadaki mm cinsinden ilerleme hızıdır. Eğer sadece koordinatların biri veya ikisi belirtilmişse hareket karşılıklı olarak tek veya iki boyutludur. Bazı kontrol birimleri eşzamanlı olarak sadece iki eksen kontrol edebilir, bu

durumlarda deyim içerisinde sadece iki koordinat belirtilmelidir. Torna için aşağıdaki format kullanılır;

G01XxZzFf

(mutlak değerlerle programlama için)

G01UuWwFf

(artışlı değerlerle programlama için)

G01 kodu belirtildiğinde NC kontrol birimi, takımını veya tablayı hareket ettiren üçmotor için üç adet sinyal seti üretir. Bu üç motor için üretilen sinyallerin saniyedeki sayısının oranı, sonuçta takım hareketinin başlangıç ve bitiş noktalarını birbirine düz olarak bağlayan bir çizgi halinde olmasını ve takımın F kodu ile belirtilen hızın aynıysa hareket etmesini sağlar. F kodu eğer bir önceki deyimdeki hızla aynıysa ise kullanılabilir.

G01 kodunda kesici yolu devamlı olarak kontrol altında olduğu için bir kontur hareketidir. Normal olarak doğrusal bir profilli işleyen takım yolunu tanımlamak için kullanılır.

Dairesel Enterpolasyon Hareketi. NC'li frezelerin ve tornaların çoğu takımını X-Y, Y-X veya X-Z düzlemlerinde daireSEL bir yol boyunca hareket ettirebilecek kapasiteye sahiptir, bu işlem G02 ve G03 daireSEL enterpolasyon kodları tarafından aşağıdaki formatla belirtilebilir;

- X-Y düzleminde;

G17G $\begin{Bmatrix} 02 \\ 03 \end{Bmatrix}$ XxYy $\begin{Bmatrix} Rr \\ #Jj \end{Bmatrix}$ Ff;

- Y-Z düzleminde;

G19G $\begin{Bmatrix} 02 \\ 03 \end{Bmatrix}$ YyZz $\begin{Bmatrix} Rr \\ #Kk \end{Bmatrix}$ Ff;

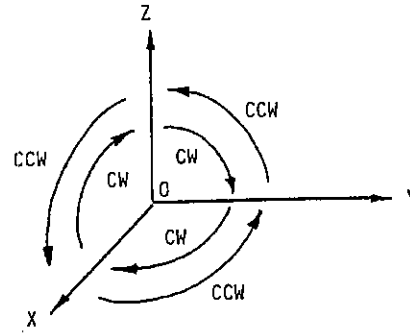
- Z-X düzleminde;

G18G $\begin{Bmatrix} 02 \\ 03 \end{Bmatrix}$ XxZz $\begin{Bmatrix} Rr \\ #Kk \end{Bmatrix}$ Ff;

burada;

G17,G18 ve G19 karşılıklı olarak daireSEL hareketlerin gerçekleştirildiği X-Y,Z-X ve Y-Z düzlemlerini tanımlarlar.

G02 ve G03 karşılıklı olarak saat yönünde (CW) ve saatin tersi yönde (CCW) daireSEL hareketleri tanımlarlar (Şekil 2-10). Saat yönünde ve saatin ters yönünde dönmeler, daireSEL hareketin gerçekleştirileceği düzleme dik olan koordinat eksenine pozitif yönden bakılması kuralınada yanarak ayırt edilebilir ve hareket karşılıklı olarak saatin veya saatin tersi yönündedir.



Şekil 2-10 Dairesel enterpolasyon hareketinin saatın yönünde (CW) ve saatın ters yönünde (CCW) tanımlanması.

x, y ve z varış noktasının koordinatlarını ya G90 ile mutlak ya da G91 ile artışı değerleri ile gösterir. Artışlı programlamada, torna tezgahı için X ve Z kodları U ve W kodlarına değiştirilmelidir.

i, j ve k dairenin merkezinden takımın ilk konumuna olan artışı koordinatları belirtir.

r dairenin yarıçapını gösterir. Eğer dairesel kesici yolu 180 dereceden büyükse, r negatif olmalıdır. Tornalama işleminin özelliğinden dolayı, torna için daireysel hareket sadece 180 dereceden az olabilir.

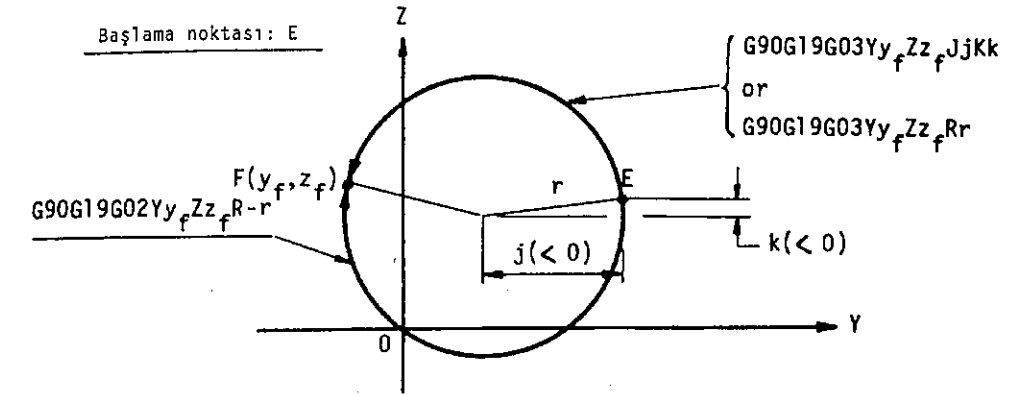
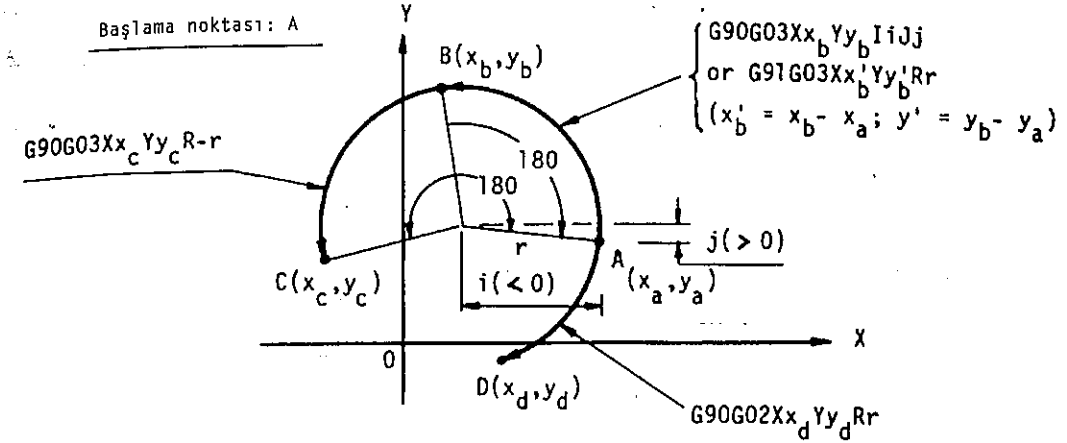
G02 ve G03 kodlarının kullanımı ile ilgili örnekler Şekil 2-11'de gösterilmiştir. NC programlamada çok sık olarak i, j ve k değişkenlerinin değerlerinde ve işaretinde hatalarla karşılaşmaktadır. Çizimlerde genellikle yarıçap verildiği için I, J ve K kodlarının yerine R kodunun kullanılması halinde hata yapılması daha az olacaktır. Bir daireysel enterpolasyon programlandığında, eğer son noktanın koordinatları başlangıç noktasının koordinatları ile aynı ise X, Y ve Z kodları kullanılmayabilir. I, J ve K kodlarını kullanmamak dairenin merkezinden X, Y ve Z eksenindeki başlangıç noktasına olan artışı koordinatın sıfır olduğu anlamına gelir. Şekil 2-12 bir veya iki kodun kullanılmadığı daireysel yolların programlanması ile ilgili bir çok örneği göstermektedir. Tam bir daire programlandığında aşağıdaki formatlarda hazırlanmış deyimler kullanılabilir;

$$G \begin{Bmatrix} 02 \\ 03 \end{Bmatrix} G \begin{Bmatrix} I \\ J \\ K \end{Bmatrix}$$

Bununla beraber, yukarıdaki deyimlerde eğer I, J ve K kodları yerine R kodu kullanılırsa hiç bir hareket oluşmaz.

Bulunulan Konumda Bekleme. Aşağıdaki formatların biri ile belirtilen G04 kodu;

$$G04 \begin{Bmatrix} X_n \\ P_n \\ U_n \text{ (for lathe only)} \end{Bmatrix}$$



Freze

G → H (CW):
G90G18G02X_h Z_h R_r

or
G91G18G02X_u Z_w R_r

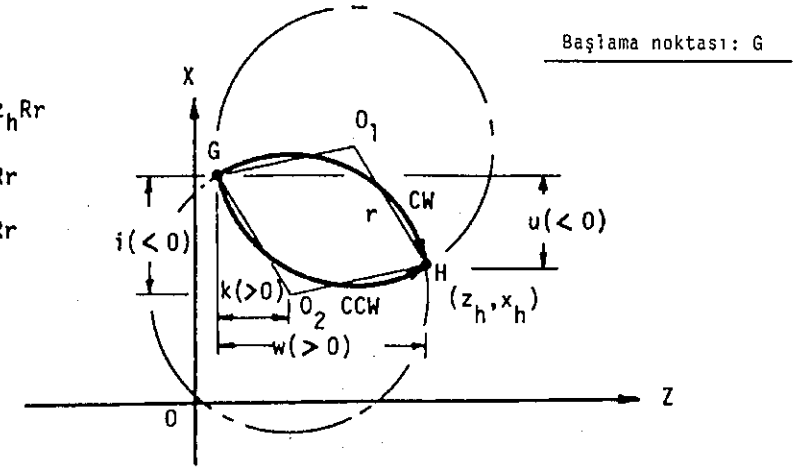
G → H (CCW):
G91G18G03X_u Z_w R_r

Torna

G → H (CW):
G02X2x_h Z_z_h R_r

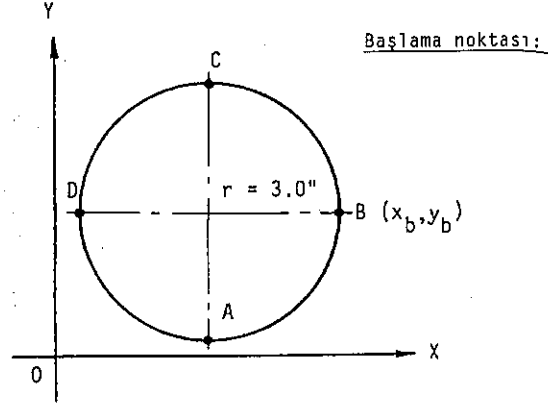
or
G02U2u W_w R_r

G → H (CCW):
G03U2u W_w I_i K_k



Şekil 2-11 Değişik koordinat düzlemlerinde G02 ve G03 kodlarının kullanımı

A → B (CCW):
G90G03X_bY_bJ3.0
or
G91G03X3.0Y3.0J3.0
A → C (CW):
G91G02Y6.0J3.0
B → D (CCW):
G91G03X-6.0R3.0



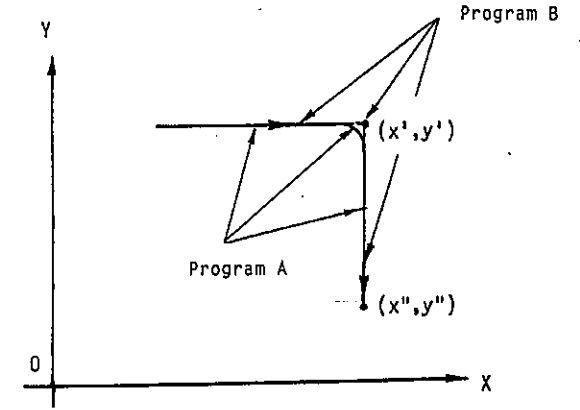
Şekil 2-12 Dairesel interpolasyon hareketini programlamada bazı kodların çıkarılabileceğini gösteren örnekler.

G04 {
X_n
P_n
Un (yalnız tornalamada) }

takımı bulunulan konumda n saniye kadar durdurmak için kullanılabilir. Bu kodun kullanılış yerlerinden birisi de parçanın profili üzerinde keskin köşe elde etmektir. Kesici hareketi esnasında, bir deyim tarafından verilen hareketin sonunda yavaşlamak ve sonraki deyim başında hızlanma olayı otomatik olarak NC kontrol birimi tarafından gerçekleştirilir (Şekil 2-13). Böylece programlanmış kesici yolları köşeli bile olsa yuvarlak bir köşe elde edilir. Keskin bir köşe elde etmek için iki deyim arasına bir G04 kodu eklenebilir.

2.6.2 Çevresel İşleme Hareketinin Tanımlanmasında Kesici Yarıçapının Telifisi

Önceki bölümlerdeki deyimler, kesicinin uç yarıçapı merkezini tanımlamak için kullanılmaktadır. Şekil 2-1'de gösterilen örnekteki X-Y düzlemindeki kesici izleri parça profilinin koordinatlarına (1.0,1.0), (1.0,-1.0), (-1.0,1.0) ve (-1.0,-1.0) dayanarak programlanamaz. Esasında takım doğru profili üretebilmek için 0.375 inç dışarı kaydırılmalıdır. Tornalama işlemlerinde konumlar ve kesici yolları takımın uç merkezine ait deyimler tarafından belirtilir. Bundan dolayı, çevresel işleme hareketleri parça resmi üzerinde belirlenen boyutlara bağlı olarak doğrudan tanımlanamaz. Bir NC programının her çevresel işleme hareketi deyiminin sonunda takımın konumunun koordinatlarının hesaplanması gereklidir. Bu hesaplama zaman alır ve hata yapılmaya müsaittir. Modern CNC tezgahlarında, özel hesaplama



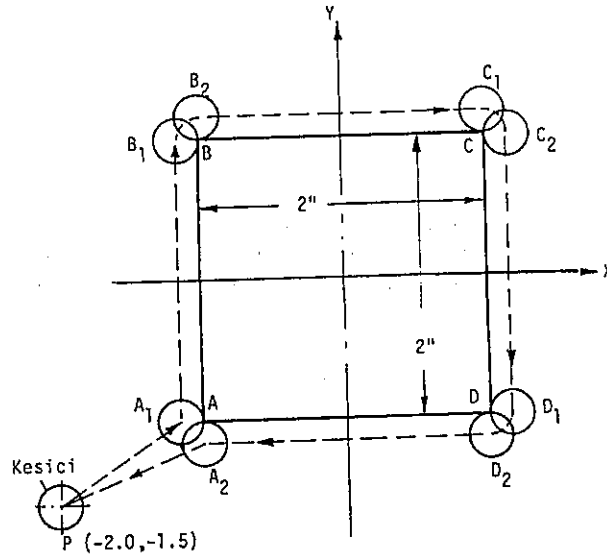
Program A	Program B
.....
G01Xx'Yy'Ff;	G01Xx'Yy'Ff;
Yy'';	G04X0.1;
.....	G01Yy'';
.....

Şekil 2-13 Bir deyim tarafından verilen hareketin sonunda yavaşlamak ve sonraki deyim başında hızlanma olayı otomatik olarak NC kontrol birimi tarafından gerçekleştirildiğinden, sonuçta yuvarlak bir köşe elde edilir. Keskin köşe elde etmek için iki blok arasına bir G04 eklenebilir.

fonksiyonları veya kesici-yarıçapı telifisi kodları ile kullanıcının parça çiziminden çevresel işleme hareketi için program yapması amacı ile hazırlanmıştır. Bunlar karşılıklı olarak bir profilin soldan takım yarıçapı telifisi ve sağdan takım yarıçapı telifisi olan G41 ve G42 kodlarıdır. Sağdan veya soldan kesici yarıçapı telifisi takımın programda belirlenen çevresel işleme hareketi yönünde profil boyunca giderken takımın sol tarafta veya sağ tarafta olmasıdır. G40 kodu kesici telifisini iptal etmek için kullanılır. Kesici telifisi kodları ile beraber yarıçap telifisi kodu olan D değeri de belirtilmelidir. Bir NC programında çevresel işleme hareketine başlamak için (takımı bulunduğu konumdan çevresel işleme hareketi için hazır bir konuma getirmek için) G41 ve G42 kodları belirlenmelidir. G41 ve G42 kodları aynı zamanda kontrol birimine takip eden deyimdeki hareketin iptal edilene kadar takım-yarıçapı telifisi modunda tanımlandığını bildirir. Örneğin Şekil 2-14'de Şekil 2-1'de gösterilen parçanın üst görünüşü belirtilmiştir ve istenilen profil ABCD'dir. Aşağıdaki program gerekli kesicinin, çevresel işleme hareketini tanımlar;

.....
 G90G20.....

 N0010G17G41G01X-1.0Y-1.0D01F2.0; (P'den A₁'e başlangıç hareketi)
 N0015Y1.0;
 N0020X1.0; (kesici yarıçapı telafisi
 modunda çevresel işleme hareketi deyimleri)
 N0025Y-1.0;
 N0030X-1.0;
 N0040G40G00X-2.0,Y-1.5; (takımı P noktasına hareket ettir ve kesici.
 yarıçapı telafisini iptal et.)



Şekil 2-14. Kesici yarıçapı telafisi kodunun bir çevresel işleme hareketini programlamak için kullanılması.

N0010 ile etiketlenmiş deyim kesici yarıçapı telafisinin soldan olduğu çevresel işleme hareketinin başlangıcını tanımlar. Kayma değerlerinin sayıları elle giriş yapılarak kontrol sisteminin belleğine saklanabilir (No 1,2 kayma kaydedicilerine). D01 kodu No 1'de saklanmış yarıçap kaymasının kesici telafisinde kullanılmaması gerektiğini belirtir. Kayma değerini girmenin bir yolu da G10 kodunu aşağıdaki deyim formatında kullanmaktır;

G10PpRr

Burada G10 kodu No p kaydedicisine r telafi değerinin giriş yapılmasına neden olur. Bu deyim belli olduğu üzere bir D kodundan önce belirtilmelidir. Bu durumda 0.375 inçlik bir kayma değerini belirten deyim şudur;

G10P01R0.375;

Yukarıdaki programda N0010 deyiminde kullanılan D01 kodu kesici yarıçapı-telafisi olarak No 1 kaydedicisindeki yarıçap kaymasının kullanılması gerektiği anlamına gelir. N0010 etiketli deyim şunları belirtir;

- Başlangıç hareketi 2.0 inç/dak ilerleme hızına sahip doğrusal enterpolasyonlu bir hareket olmalıdır.
- No 1 kayma kaydedicisindeki saklanan değer tarafından belirlenen kesici yarıçapı telafisi sol tarafta olmalıdır.

Telafisinin yönü (sağdan veya soldan) iki deyim tarafından belirlenir: başlangıç hareketi deyimini (N0010 etiketlenmiş) ve hemen ardından gelen deyim (N0015 etiketlenmiş). Başlangıç hareketinin sonunda, takım bir sonraki hareket deyimini tarafından tanımlanmış istenilen yolu teğet olacağı konuma hareket etmek zorundadır. Takibeden deyimlerdeki kesici yolu parça profiline göre tanımlanır. N0040 olarak etiketlenmiş deyim A₂ noktasından P noktasına doğru kesici telafisini yavaşça kaldırır.

Kesici yarıçapı telafisi çevresel işleme hareketlerini tanımlanması sırasında karşılaşılabilecek ve yeni başlayanları çelişkiye düşürebilecek değişik durumlar aşağıda açıklanmıştır.

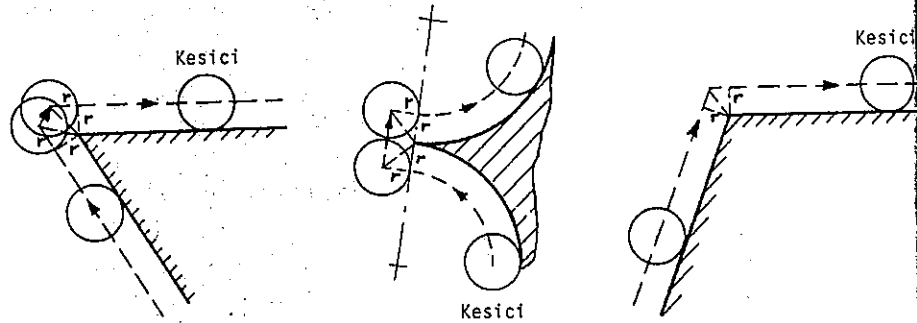
1. G41 ve G42 kodları ile tanımlanan telafinin yönü şu şekilde tanımlanır: bir profil boyunca bir noktadan diğerine belirlenen yönde giderken karşılıklı olarak birinde takım solda diğerinde ise sağda bulunur. Yukarıda gösterilen programda A noktasından B noktası ile C,D ve A noktasına giderken takım sol tarafta olmalıdır.
2. Başlangıç hareketi deyimlerinde sadece G00 ve G01 kullanılabilir.
3. Başlangıç hareketi deyiminden hemen sonra gelen deyim belirlenen başlangıç hareketi deyiminin düzleminde bir hareketi tanımlamalıdır. Eğer bir hareketi tanımlayan veya başlangıç hareketinin tanımlandığı düzlemdeki hareketten başka bir hareket belirtilirse kontrol birimi telafisinin yönünü saptamayı başaramayacaktır. Çevresel işleme hareketinin ikinci deyiminden sonra her G41 ve G42 modundaki her deyim için telafisinin yönü takip eden iki deyim referans alınarak saptanabilir. Bundan dolayı hareket içermeyen bir deyim veya kesici yarıçapı düzleminde olmayan bir hareketi tanımlayan bir deyim araya sokula-

bilir. Örnek olarak aşağıdaki deyim;

N0018Z-0.5;

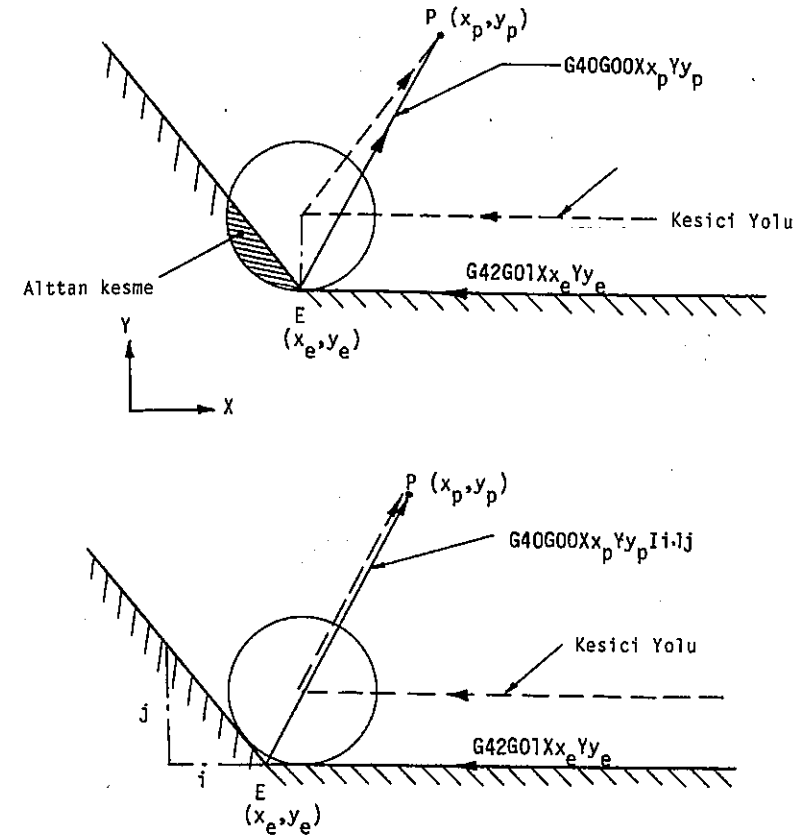
N0015 ve N0020 deyimleri arasına takımın 0.5 in aşağıya hareket ettirmek için sokulabilir ve kesici çevresel işleme hareketi bundan sonra X-Y düzleminde devam edecektir. NC kontrol birimi, iki hareket içermeyen deyim veya telafi düzleminde olmayan iki hareket deyimini peşpeşe tekrarlanırsa telafi yönünü bulamayacaktır.

4. Başlangıç hareketinin ve sonraki deyimın kesici yolları arasında kalan açı 90 dereceye eşit veya büyük olmalıdır. Bunun dışında başlangıç hareketi gerçekleşmeyecektir.
5. FANUC 6MB kontrol birimi için bir profilin köşesindeki takım hareketi kontrol birimini tasarımına ve parametrelerinin ayarlanmasına bağlıdır ve Şekil 2-15'te gösterilen formda olabilir.
6. G00 ve G01 kodları ile beraber en son deyimde (N0040 etiketlenmiş) kullanılan G40 kodu kesici telafisini bu satırın başlangıcından sonuna doğru yavaşça iptal eder. Bu deyim için takımın ilk konumunun bir önceki deyimın son noktası olduğuna ve takımın parça profiline teğet olduğuna dikkat edin. Sonuçta bazı koşullarda (Şekil 216) istenmeyen bir dalma oluşabilir. Telafinin uygulandığı düzleme bağlı olarak, iki kod ya I ve J, J ve K veya I ve K kodları, dalmanın engellenmesinin istendiği parça profilini belirlemek için son deyimlere eklenebilir (Şekil 2-16).



Şekil 2-15 Değişik iş parçalarının köşelerinde kesici hareketleri

Bir torna için kesici telafisi aşağıdaki nedenlerden dolayı bir freze için olandan daha karmaşıktır. Bir tornanın kesicisinin ucu bir dairenin sadece bir parçasıdır ve parmak freze gibi işleme hareketi esnasında dönmeyiz. Bu yüzden farklı kesici telafi vektörleri (veya yönleri) farklı kesici takımlar için uygulanmalıdır (Şekil 2-17). Ek olarak döner başlığa bağlanmış takımlar başlık merkezine göre farklı konumlara sahiptirler. Bu farklılıkları telafi etmek için X ve Z kaymalarını farklı takımlar için



Şekil 2-16 Son kesici telafi hareketinde dalma oluşabilir; I ve J son deyimler bu davranışı engellemek için eklenir.

ayarlanmak zorundadır. Bundan dolayı X ve Z kaymaları, takım ucu yarıçapı r ve takım vektörünün numaraları takım kayma kaydedicilerine program işletilmeden evvel yazılmalıdır. FANUC6T kontrol sistemi için yukarıdaki verileri kayma kaydedicilerine koymak için G10 kodu aynı şekilde kullanılabilir. FANUC6T kontrol sisteminde takım kayma görünümünün bir örneği Tablo 27'de gösterilmiştir.

2.7 İŞLEME ÇEVİRİMLERİNİ VE TEKRARLAMALI İŞLEME ÇEVİRİMLERİNİ TANIMLAYAN KODLAR

Torna ve freze tezgahları üzerindeki tezgaah işlemleri belli bir sıradadır ve doğaldır. Bu nedenle çevrimseldir. Örneğin aşağıdaki takım hareketleri verilen konumdaki deliği delmek için gerçekleştirilmelidir;

- Takımı istenen konuma hızlı hareket ettirme
- Takımı parça yüzeyinin çok az üzerine (açıklığa) hızlı olarak indirme
- Takımı istenilen derinliğe programlanmış ilerleme hızı ile hareket ettirme (deliği delme)
- Takımı yukarı istenen konuma kaldırma

Bundan dolayı delme işlemi için bir alt yordam yazılabilir. Alt yordamı çağırırken bir kullanıcı sadece istenilen konumun koordinatları, delme derinliği ve açıklık miktarı olan uygun parametreleri atamak zorundadır.

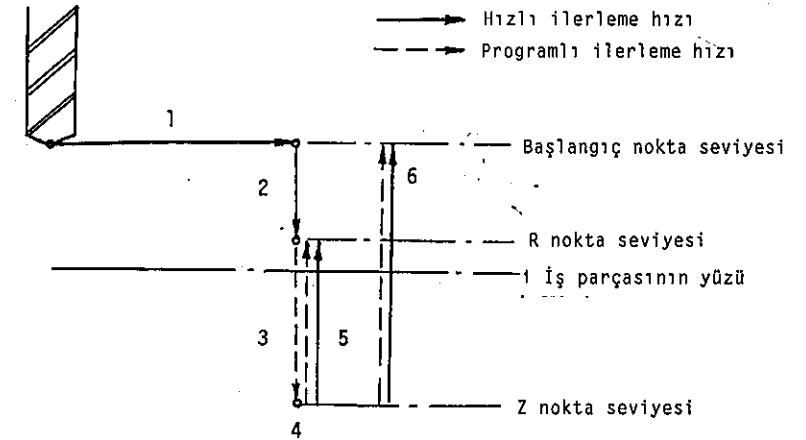
FANUC6MB ve 6T kontrol sistemlerinde delme ve delik işleme gibi farklı tezgaah işlemlerini gerçekleştirmek için bir çok alt yordam kullanılabilir ve çalıştırmak için saklanabilir. Bu saklanmış alt yordamları çağırmak ve işletmek için kodların özel bir kombinasyonu gereklidir.

2.7.1 Freze Tezgaahlarında Talaş Kaldırma Çevrimleri

Özel G kodlarını kullanan deyimlerle programlanabilecek bir çok talaş kaldırma çevrimi vardır. Bunlar delme, kademeli delme, dış çekme ve delik işleme çevrimleridir. Bu çevrimlerin programlanması için kullanılan yaygın kodlar aşağıdakilerdir;

G80	Çevrim fonksiyonunu iptal
G81	Delme çevrimi
G82	Havşa açma çevrimi
G83	Boşaltmalı delme (Kademeli delme) çevrimi
G84	Kılavuz çekme çevrimi
G85	Delik büyütme çevrimi
G86'dan G89'a	Özel delik büyütme çevrimleri

Tezgaah çalıştırıldığında NC kontrol birimi G80 modundadır (G80 kodu etkilidir). Bir talaş kaldırma çevrimi G81,... veya G89 kodu ile programlanabilir ve G80 kodu veya Grup 01'de yer alan herhangi bir G kodu ile iptal edilebilir.



- İşlem 1: Takımın istenen konuma hızlı hareketi
 2: Takımın parça yüzeyinin üzerinde istenilen açıklıkta R noktasına hızlı inişi
 3: Takımı programlanmış ilerleme hızında aşağıya hareket ettirme (delmeyi, delik işi büyütmeyi gerçekleştirme)
 4: Dipte gerçekleştirilen işlem (iş milini durdurma, bekleme vs.)
 5: R noktasına hızlı veya programlanmış hızda dönme (G99 modu)
 6: İlk noktanın seviyesine hızlı veya programlanmış hızda dönme (G98 modu)

Şekil 2-19 Bir çevrimdeki (freze tezgahu) işlemler.

Yukarıdaki kodlarla belirlenen işlemler genellikle Şekil 219'da gösterilen işlemlerin bazılarını veya hepsini kapsayabilir. Takım önce yatayda hızlı olarak X,Y kodları ile belirlenen noktaya hareket eder, oradan hızlı olarak R kodu tarafından belirlenen seviyeye iner. Sonra takım programlanmış ilerleme hızı ile Z kodu tarafından belirlenen seviyeye hareket eder. İki türlü geri çekme hareketi vardır; bunlar G89 veya G99 kodunun G81, 82 veya G89 kodunu saptayan deyimden önce belirlenmesi ile seçilir. G89 kodu etkili olduğunda takım ilk noktanın seviyesine döner G99 modunda takım R noktası seviyesine geri çekilir.

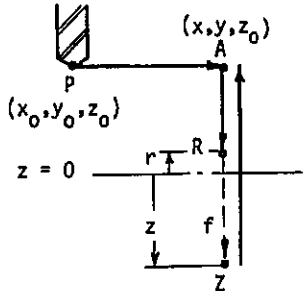
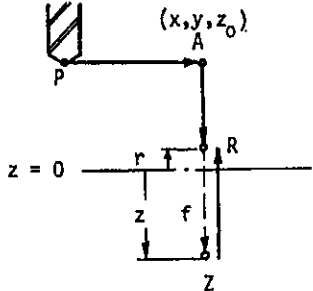
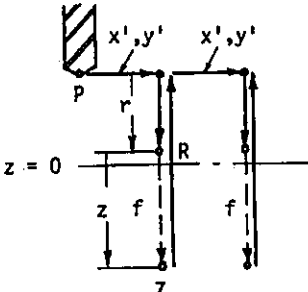
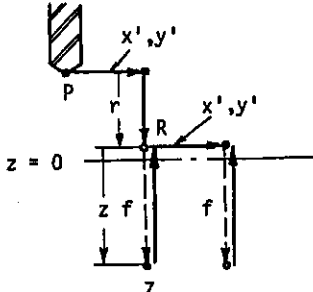
Talaş kaldırma çevrimleri G90 ve G91 modunda da programlanabilir. R ve Z kodlarının tanımları bu iki modda farklıdır.

G81 Kodu (Delme) G81 kodu saklanmış delme çevrimi alt yordamını çağırmak için kullanılır, X,Y,Z,R,F ve L kodları gerekli parametreleri atamak için kullanılmalıdır. Delme çevrimini tanımlayan bir deyim formatı şu şekildedir:

G81XxYyZzRrFfLl';

burada (Şekil 2-20);

x ve y = delinecek deliğin koordinatları. G91 modunda x ve y , bulunan takım konumundan delik konumuna olan artışlı koordinatlarıdır.

	G98	G99
G90	 <p>G81XxYyZzRrFf;</p>	 <p>G81XxYyZzRrFf;</p>
G91	 <p>G81Xx'Yy'ZzRrFfL2;</p>	 <p>G81Xx'Yy'ZzRrFfL2;</p>

Şekil 2-20 G81 delme çevrimi tarafından tanımlanan takım hareketi ve G81 kodu ile birlikte kullanılan kodların anlamı.

z = delik dibinin koordinatı. G91 kodu etkili ise R noktasından olan artışlı koordinattır.

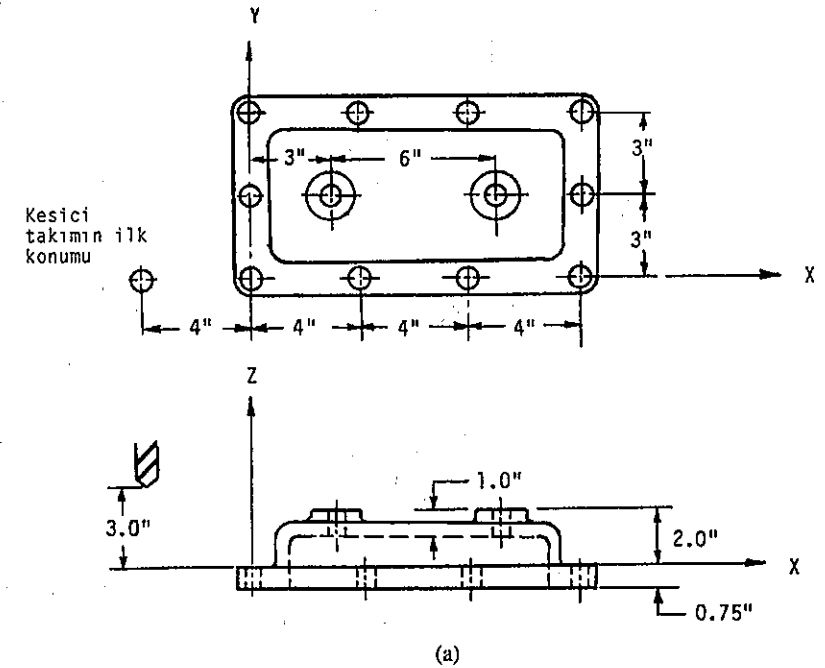
r = R noktasının koordinatı (G90 modunda) veya ilk noktadan P noktasına artışlı koordinat (G91 modunda).

f = dakikadaki mm veya inç biriminden ilerleme hızı (miktarı)

l = Bu altıyordamın kaç defa çalıştırılacağı sayısını. G90 modunda, L kodunun delme işlemine etkisi yoktur. Takım (x, y) konumunda L kere dalıp çıkacaktır. G91 modunda bir L kodu birbirleri arasında eşit mesafeler olan L sayıda delik oluşturacaktır. Eğer L kodu deyimde kullanılmazsa onun yerine varsayılan değer kullanılır.

Bu deyim tarafından farklı modlarda kesici hareketleri Şekil 2-20'de gösterilmiştir.

Şekil 2-21 (a)'da gösterilen parça delme çevrimini açıklamak için örnek olarak kullanılmıştır. İlk olarak $(-4.0, 0.3, 0)$ konumunda olan takım ile 12 delik delmek için kullanılan program aşağıdaki gibi olabilir (Şekil 2-21(b)).



Şekil 2-21 (a) NC'li freze tezgahında delinecek 12 delikli parça. (b) NC programı (00001) tarafından tarif edilen takım yolu. Takımın (matkabın) başlangıç konumu P1'dir (deyim N1). Farklı deyimler tarafından belirlenen takım yolları aşağıdaki şekildedir; deyim N2: P1-P2-...-P6 ($y=0$ düzleminde) deyim N3: P6-P7-P8 ($x=12$ düzleminde) deyim N4: P8-P9-P10-P11 ($x=0$ düzleminde) deyim N5: P11-P12 ($x=0$ düzleminde) deyim N6: P12-P13 ($x=0$ düzleminde) deyim N7: P13-P14-P15 ($Y=3$ düzleminde) deyim N8: P15-P16 ($y=3$ düzleminde) deyim N9: P16-P1

- q = Herbir delme kademesi için kesme derinliği (daima pozitif bir artış değeridir)
 f = dakikada inç veya milimetre cinsinden ilerleme hızı
 l' = bu çevrimin kaç defa tekrarlanacağını gösteren sayı

Bu deyim tarafından tanımlanan hareket Şekil 2-22'de gösterilmektedir. Matkapın, aşağı hareket etmekte iken, bir önceki talaşın bitiş konumundan d inç kadar yukarıda olan bir konumda yüksek hızdan programlanmış ilerleme hızına geçtiğine dikkat edilmelidir. d 'nin değeri, NC denetleyicisinin parametre ayarına göre belirlenir. G83 kodunun kullanımı, esas itibarıyla G81 kodunki ile aynıdır.

	G98	G99
G90	<p>G83XxYyZzQqRrFf;</p>	<p>G83XxYyZzQqRrFf;</p>
G91	<p>G83Xx'Yy'ZzRrQqFfL2;</p>	<p>G83Xx'Yy'ZzQqRrFfL2;</p>

Not : ————— Hızlı hareket; - - - - - programlanmış besleme hızındaki matkap hareketi

Şekil 2-22 G83 koduyla tanımlanan kademeli delme çevriminin kesici hareketi.

G85 Kodu (Delik Büyütme Çevrimi). Delik büyütme işlemi, kesici takımın R ve Z noktaları arasında iken, programlanmış bir ilerleme hızında hareket etmesini gerektirir. Delik büyütme çevrimi altyordamının çağrılmasını sağlayan deyim:

G85XxYyZzRrFfLl';

olup; buradaki x , y , z , r , f ve l' kodları delik delme çevrimi için kullanılan kodlarla (G81) aynı anlamları ifade eder. G81 ve G85 kodları arasındaki tek fark, geri çekme hareketidir (Şekil 2-23). G85 kodu, G81 kodu ile aynı biçimde kullanılabilir.

	G98	G99
G90	<p>G85XxYyZzRrFf;</p>	<p>G85XxYyZzRrFf;</p>
G91	<p>G85Xx'Yy'ZzRrFfL2;</p>	<p>G85Xx'Yy'ZzRrFfL2;</p>

Not: ————— Hızlı hareket; - - - - - programlanmış besleme hızındaki kesim hareketi

Şekil 2-23 G85 koduyla tanımlanan bir delme çevriminde kesici takım hareketi.

Diğer G80 kodlarının işlevi, yukarıda incelenmiş olan üç kodun işlevlerine benzer. Bu konuda ayrıntılı bilgi edinmek için, okuyucunun, kullanılan NC tezgahın programlama kılavuzuna (örneğin 26 numaralı referans) başvurması gerekir.

2.7.2 Bir Torna Tezgahındaki Talaş Kaldırma Çevrimleri

Bir tornadaki bir kesme çevrimi genellikle, (delik delme çevrimi hariç) X ve Z doğrultularının her ikisinde de hareket gerektirir ve esas itibarıyla Z doğrultusunda kesme hareketi gerektiren bir freze tezgahından farklıdır. FANUC 6T denetleyicili bir NC'li torna tezgahında, sırasıyla, bir tek çevrim programlamak ve (bir pasoda alınamayan fazla malzemeyi almak için için kullanılan) tekrarlamalı bir çevrim programlamak amacıyla olmak üzere; iki cins kod kullanılabilir.

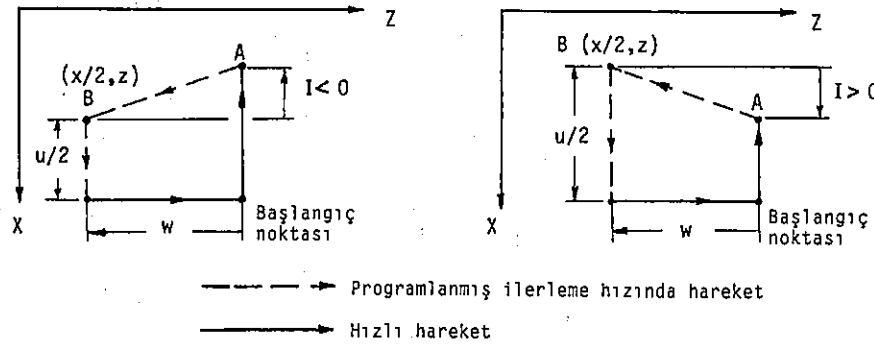
G90 Kodu (Silindirik veya Konik Yüzey İşleme Çevrimi). Bu çevrimi tanımlayan deyimnin genel formatı:

$$G90 \left\{ \begin{array}{l} XxZz \\ UuVvWw \end{array} \right\} I f \#$$

burada:

$x/2$ ve z = varış noktasının koordinatları

$u/2$ ve w = başlangıç noktasından varış noktası (B)'ye kadar olan artışlı koordinatlar



Şekil 2-24 G90 (torna) koduyla tanımlanan çevrim hareketi.

i = B varış noktasından A noktasına kadar ölçülen artışlı koordinatlar (Eğer $i = 0$ ise, ihmal edilebilir)

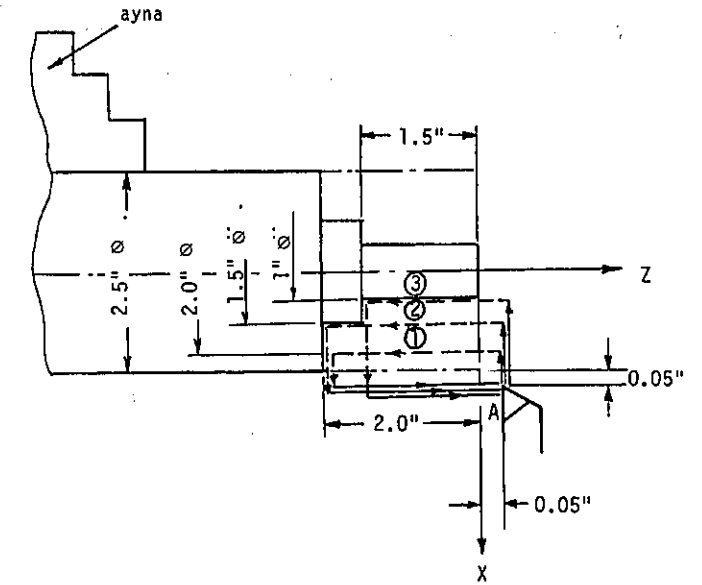
f = Devir başına inç veya milimetre cinsinden ilerleme hızı

* = FANUC 6T denetleyici için blok sonu işareti

Bu çevrimdeki birinci ve üçüncü hareketlerin sırasıyla hızlı ve programlanmış ilerleme hızında Z eksenine dik olduğuna dikkat edilmelidir. Son hareket ise Z eksenine paralel doğrultudadır. G90 kodunun, kesme işlemini tanımlamak için kullanımını gösteren bir Örnek Şekil 2-25'de verilmiştir.

G94 Kodu (Alın İşleme Çevrimi). İşlevi iş parçasının çapını düşürmek olan ve bir G90 koduyla tanımlanan tornalama işleminin aksine; alın işleme, tornalanmış bir parçanın boyunu düşürmek için kullanılan bir işlemdir. Bir alın işleme çevrimini tanımlayan deyimnin genel formatı (Şekil 2-26):

$$G94 \left\{ \begin{array}{l} XxZz \\ UuVvWw \end{array} \right\} K f \#$$



Mutlak koordinatlarda programlama

M21 (torna kalem A konumundadır)
N22G90X2.0Z-2.0F0.01* (1. çevrim)
N23X1.5* (2. çevrim)
N24X1.0Z-1.5* (3. çevrim)
.....

Artışlı koordinatlarda programlama

M21 (torna kalem A konumundadır)
N22G90U-0.6W-2.05F0.01* (1. çevrim)
N23U-1.1* (2. çevrim)
N24U-1.6W-1.55* (3. çevrim)
.....

Şekil 2-25 G90 kodu kullanılarak yapılan bir torna programlama örneği.

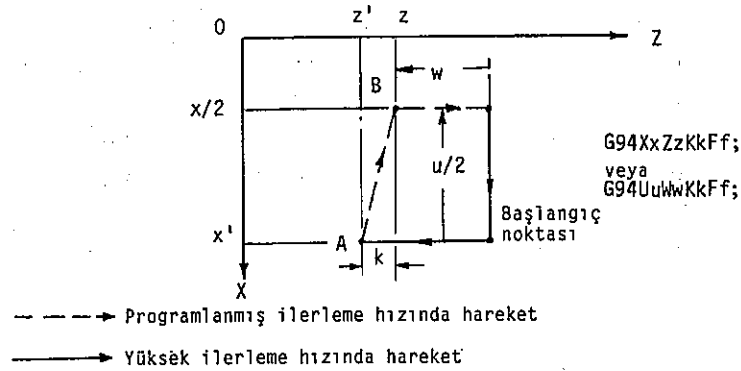
burada; x , z , u , w ve f G90 kodunda kullanılanlarla aynı anlama sahiptir; k ise B noktasından A noktasına kadar olan artışlı koordinatları belirtir. Alın işleme çevriminin hareketi, G90 çevriminin ters yöndedir. Birinci ve üçüncü kademe hareketleri Z eksenine paralel; son hareket kademesi ise Z eksenine diktir. A'dan B'ye hareket yönü; eğer A noktası B noktasına göre Z ekseninin eksi tarafında ise

$k < 0$, artı tarafında ise $k > 0$ olmak üzere bir K koduyla tanımlanabilir. A'dan B'ye hareket doğrultusu X eksenine paralel olduğu takdirde, K kodu ihmal edilebilir.

G92 Kodu (Diş Açma Çevrimi): Diş açma çevrimi, hassas bir ilerleme/devir sayısı oranı ile birlikte kesici takım ve iş parçası arasında hassas bir izafi konumuna sürekli olarak korunması suretiyle gerçekleştirilen bir tornalama çevrimidir. Bir diş açma çevrimini tanımlayan genel deyim formatı:

$$G92 \left\{ \begin{matrix} XxZz \\ UuWw \end{matrix} \right\} \text{if} \left\{ \begin{matrix} Ff \\ Ee \end{matrix} \right\} *$$

buradaki x, z, u, w, i ve f parametreleri Şekil 2-27'de belirtildiği gibidir. Ondalık noktadan sonra altı basamaklı bir girişi kabul edebilen E kodu, daha yüksek hassasiyette diş adımına ihtiyaç olduğu zaman kullanılır.



Şekil 2-26 G94 koduyla tanımlanan alın işleme çevrimi. k değerinin, B noktasından A noktasına kadar olan artışlı koordinat olduğuna dikkat ediniz. Burada, k sıfırdan küçüktür.

Diş açma işlemi hiçbir zaman tek bir pasoda gerçekleştirilemez. Bir diş tamamlamak için, birkaç pasoya ve dolayısıyla ardarda birkaç diş açma deyimine ihtiyaç vardır. Örneğin, aşağıdaki program, Şekil 2-28'de gösterilen standartta diş açacaktır. Bu standartta (M24x3) diş açmak için, şu yedi diş açma çevrimine (N31'den N37'ye kadar olan deyimlere) ihtiyaç vardır:

.....
G21.....

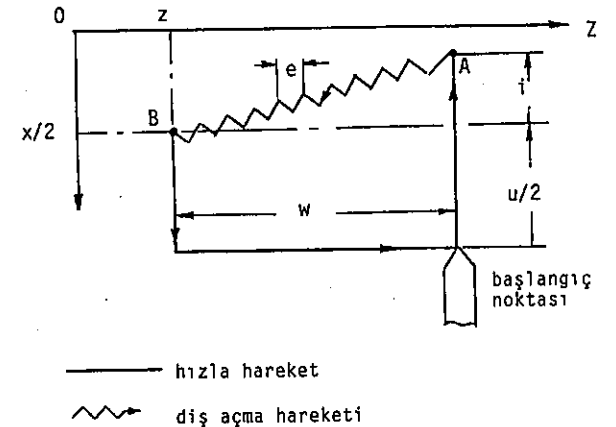
N31G92X23.0Z-40.0F3.0*
 N32X22.3*
 N33X21.8*
 N34X21.4*
 N35X21.1*
 N36X20.8*
 N37X20.752*

Denetleyici; torna kaleminin ikinciden yedinciye kadar (dahil) olan çevrimlerde yanlış bir konuma ilerlememesini sağlamak için, kalem hareketini milin hareketiyle senkronize edecektir.

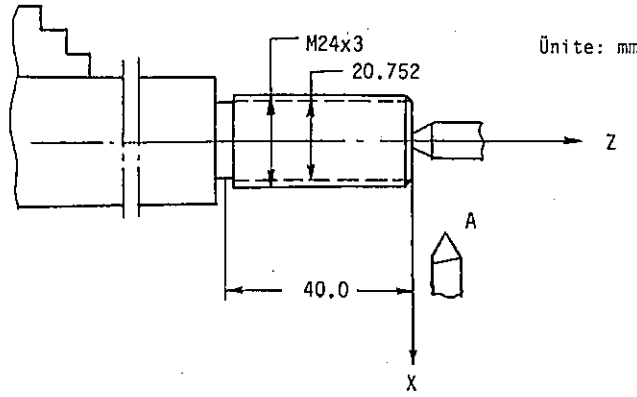
2.7.3 Torna için Tekrarlamalı Çevrim Kodları²⁷

Bir torna tezgahında yapılacak tornalama işlemi, normal olarak kaba işleme için birkaç paso ve ince (hassas) işleme için de bir paso vermeyi gerekli kılar. Ayrıca, kaba ve ince (hassas) işleme için kalemin izlemesi gereken yollar birbirine benzer veya birbirinden farklı olabilir. Torna kalemini, NC programında belirtilen kalem hareket yolunu esas alarak birkaç işleme pasosu yapacak şekilde yönlendiren bir altıyordam elde etmek mümkündür. FANUC 6T denetleyicisi sistem vasıtasıyla, bu altıyordamların bellekten çağırılması ve yürütülmesi için gerekli kodlar şöyledir:

G92XxZzIiFe veya G92UuWwiEe



Şekil 2-27 Bir G92 koduyla tanımlanan diş açma çevrimi.



Şekil 2-28 Diş açılmış parça.

- G70 İnce (hassas) işleme
- G71 Tekrarlı tornalama çevrimi (talaş kaldırma)
- G72 Tekrarlı aln işleme çevrimi (talaş kaldırma)
- G73 Dizi tekrarlama çevrimi (talaş kaldırma)
- G76 Tekrarlı diş açma çevrimi

G71 Kodu (Tekrarlı Tornalama Çevrimi). Tekrarlı talaş kaldırarak yapılacak bir tornalama işleminin tanımlanması için kullanılan format ve deyimler aşağıdaki gibidir (Şekil 2-29):

G71PpQqUuWwDdFfSsTt*

Np.....*

.....*

.....*

Nq.....*

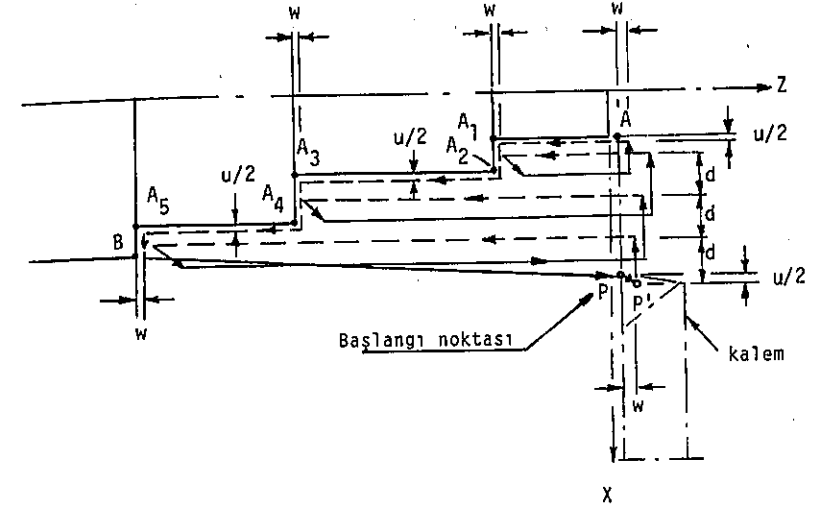
G70PpQq*

.....*

Burada;

- p = Bulunulan P konumundan A noktasına kadar olan kısımda kalem hareketini tanımlayan deyim'in sıra numarası
- q = Bu tornalama çevrimindeki son deyim'in sıra numarası (Şekil 2-29'da gösterilen durumda son deyim, A_5 noktasından B noktasına kadar olan yol boyunca kalem hareketini tanımlayan deyimdir.)

$u/2$ = X yönünde ince (hassas) işleme için çap değeri cinsinden bırakılmış olan pay



Şekil 2-29 Bir G71 koduyla belirtilmiş olan kalem hareketi: ---> programlanmış ilerleme hızı; -----> hızlı hareket.

w = Z yönünde son paso için bırakılan pay

d = Herbir pasodaki talaş derinliği

f = İlerleme hızı (Bu ilerleme hızı, p 'den q 'ya kadar işaretli deyimlerde belirtilmiş olan ilerleme hızını (veya hızlarını) hükümsüz kılacaktır.)

s = Devir sayısı (Bu devir sayısı, kalem hareket yolunu tanımlayan deyimlerde belirtilmiş olan hızı hükümsüz kılacaktır.)

t = Kalem kodu ve fark numarası

İnce (hassas) işlemin tanımlayan ve

G70PpQq*

biçiminde yazılan deyim, çevrim içindeki kalemin hareket yolunu tanımlayan son deyimden sonra belirtilebilir. Bu deyim belirtildiği anda; kalem p 'den q 'ya kadar işaretli deyimlerde belirtilmiş olan kalem hareket yolunu, devir sayısını ve ilerleme hızını esas alarak bir ince işleme yapacaktır.

NC programlamasında yaygın olarak yapılabilen bir hata; Z eksenine dik olanın dışında herhangi bir yöndeki kesici takım hareketinin, kesme çevriminin p etiketli ilk deyiminde tanımlanmasıdır. Hareketin Z eksenine dik olması gerekir. Ayrıca; çevrimin bitiş konumu veya daha belirgin bir ifade ile, q etiket deyiminde bitiş konumu B olarak tanımlanmalıdır. B noktasından P başlangıç noktasına kadar olan hareketi belirten bir deyim kullanmaya gerek yoktur. Bunun yanı sıra, kaba talaşta kalemın izlediği tüm hareket yollarının Z eksenine paralel olduğu da dikkate değer bir husustur.

Örnek

Şekil 2-29'da gösterilen işleme için gerekli kesici hareket yolu, $u = 0.01$ ve $d = 0.12$ varsayımıyla), aşağıdaki gibi programlanabilir:

....G20....

N101G71P102Q108U0.01W0.005D0.12F0.005*

N102G00X_a*

N103G01Z_{a1}*

N104X_{a2}*

N105Z_{a3}*

N106X_{a4}*

N107Z_{a5}*

N108X_{a6}*

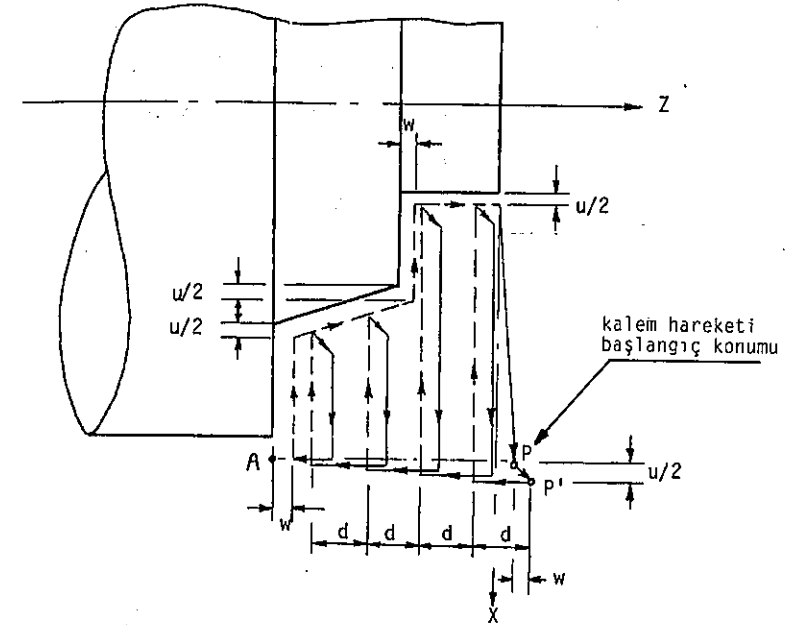
N109G70P102Q108* (Bu deyimdeki kalem hareket yolu AA₁A₂A₃A₄A₅B'dir)

G72 Kodu (Tekrarlı Alın İşleme Çevrimi). G72 kodlu deyim formatı, bir G71 kodlu deyiminkiyle aynı olmakla beraber; şu farklılıklar mevcuttur (Şekil 2-30):

1. Kalem hareketi, G71 koduyla tanımlanana ters yöndedir.
2. D kodu, Z eksen yönündeki paso derinliğini tanımlar.
3. Çevrimdeki ilk deyim, P noktasından A noktasına kadar devam eden ve Z eksenine paralel olan bir hareketi tanımlamalıdır.

G72 kodunda programlama, G71 kodundakiyle aynıdır.

G73 Kodu (Dizi Tekrarlama Çevrimi). Şekil 2-29 ve 2-30'dan görülebileceği üzere; çoklu çevrimlerin herbir pasosundaki etkin kesici takım yolları (yani, bir kesici takımın programlanmış ilerleme hızında izlediği yollar), son paso dışında, p'den q'ya kadar etiketli çevrim deyimlerinde tanımlanmış olanlarla aynı değildir.

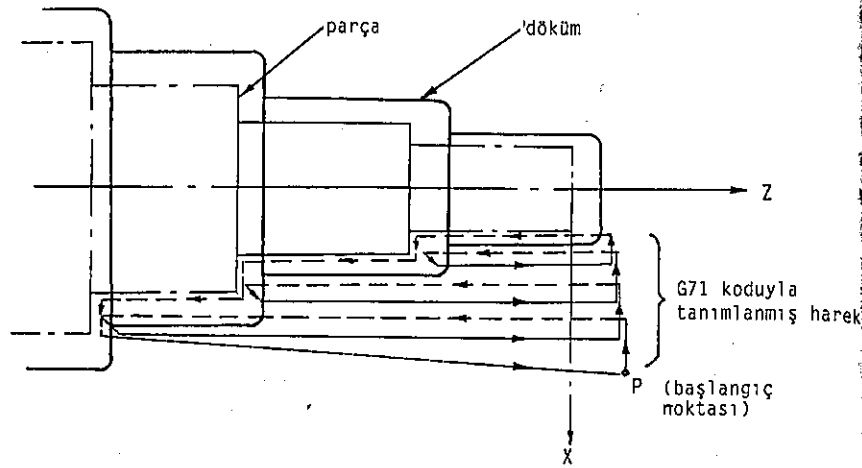


Şekil 2-30 Bir G72 koduyla tanımlanan kalem hareketi: ----> programlanmış ilerleme hızında hareket; ———> hızlı hareket.

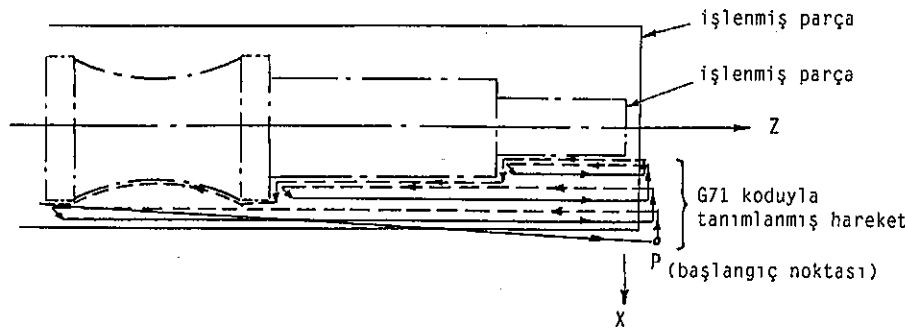
Bu yollar Z eksenine paralel (G71) ya da diktir (G72). Bir çoklu talaş kaldırma işlemini tanımlamak için bu iki kodu kullanmanın uygun olmadığı şu iki durum mevcuttur:

1. Kaba ölçülerdeki malzemenin, son işlemesi yapılmış parçanın geniş biçimindeki veya benzeri bir profile sahip olan, kalıpta dövülmüş ya da bir döküm parça olması (Şekil 2-31). Bu durumda bir G71 veya G72 kodu kullanıldığı takdirde, kesici çoğunlukla iş parçasına temas etmeyecektir.
2. Parça yüzeyinde normalin altında paso verilmesini gerektiren kısımlar bulunması (Şekil 2-32). G71 kodunda son pasonun dışında her pasodaki etkin kesici takım yolları Z eksenine paralel olduğu için, son pasoda büyük bir işleme meydana gelecektir.

Bu sorunları çözümlenebilmek için, kesici takımın p'den q'ya kadar işaretli çevrim deyimleri tarafından tanımlanmış olan dizide hareket etmesi için komut veren bir altyordamı bellekten çağırmak için bir G73 kodu kullanılabilir. Bir dizi tekrarlama çevrimini (Şekil 2-33) tanımlamak için aşağıdaki formata sahip deyimler kullanılabilir:



Şekil 2-31 Döküm veya kalıpta dövülmüş bir malzeme, G71 çok tekrarlı çevrim hareketiyle işlenemez: -----> programlanmış ilerleme hızında hareket; ——— hızlı hareket.



Şekil 2-32 Bir G71 kodu kullanıldığı takdirde, son çevrimde, normalin altında paso verilmesini gerektiren kısımda büyük bir işleme meydana gelir: -----> programlanmış ilerleme hızında hareket; ——— > hızlı hareket.

G73PpQqliKkUuWwDdFfSs*

Np.....*

.....*

.....*

Nq.....*

G70PpQq*

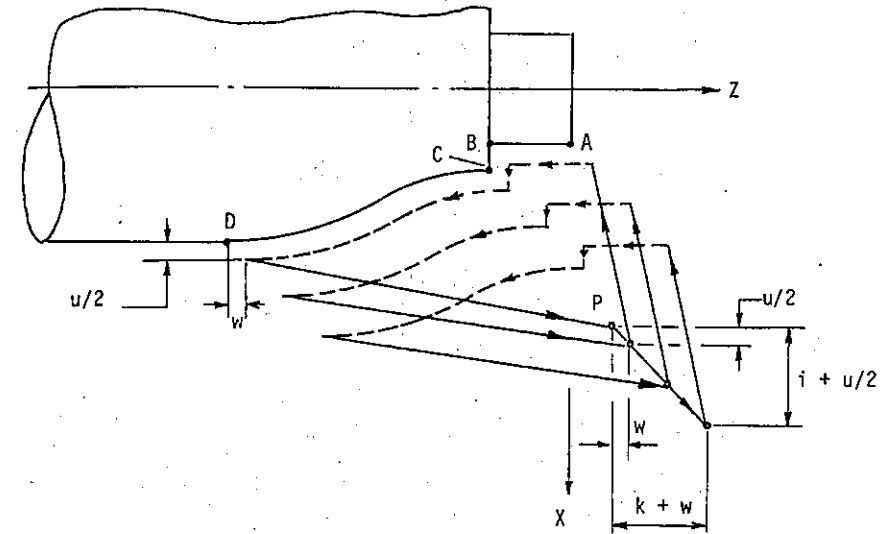
Burada;

p ve q = (Şekil 2-33'de $P \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ olarak gösterilen kesici yolunu tanımlayan, sırasıyla ilk ve son deyimlere ait deyim etiketleri

- i ve k = Çok pasolu işleme için radyal ve aksel geri çekme
- $u/2$ ve w = İşlemede son paso için izin verilen radyal ve aksel pay
- d = Tekrarlanalı paso sayısı
- f = Devir başına inç veya milimetre cinsinden ilerleme miktarı
- s = Devir sayısı

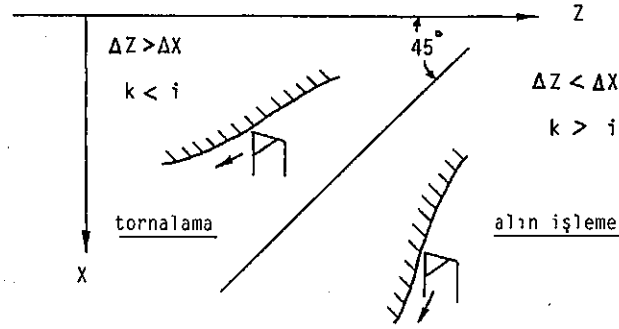
demektir.

Şekil 2-33'de görülebileceği üzere, G71 ve G72 kodlarının tersine, p etiketli deyim tarafından belirtilen hareket yönünde hiçbir sınırlama yoktur. NC programlama işinde yeni olan bir eleman, i ve k değerlerini belirlemede çoğu kez güç durumda kalacaktır. Aslında i ve k değerleri, çoklu pasolarda, X ve Z yönlerindeki toplam paso verme miktarını ifade eder. Bu değerler, çevrimdeki ilk ve son kesici yollarını önceden tanımlamak suretiyle kolayca belirlenebilir.



Şekil 2-33 Bir G73 koduyla tanımlanan dizi tekrarlayan çevrimde kesici hareketi: — > programlanmış ilerleme hızında hareket; ——— hızlı hareket.

Bir tekrarlanalı dizisi esas itibarıyla bir tornalama işlemi olduğu takdirde, Z yönündeki hareket miktarı X yönündekine kıyasla daha büyüktür (Şekil 2-34'de ΔZ ve bu nedenle de k 'nin i 'den daha küçük olması gerekir. Aksi takdirde $k > i$ 'dir. Bir deyimde I ve K kodlarının her ikisini birden belirtmek gerekli olmamasına rağmen, NC programcısı, değişik pasolarda işlenen malzeme miktarlarının mümkün olduğu kadar eşit olmasını sağlamak için I ve K kodlarının doğru birleşimini seçmelidir. Eğer I ve K kodlarından biri ihmal edilmişse, NC denetleyicisi tarafından sıfır olarak dikkate alınır.



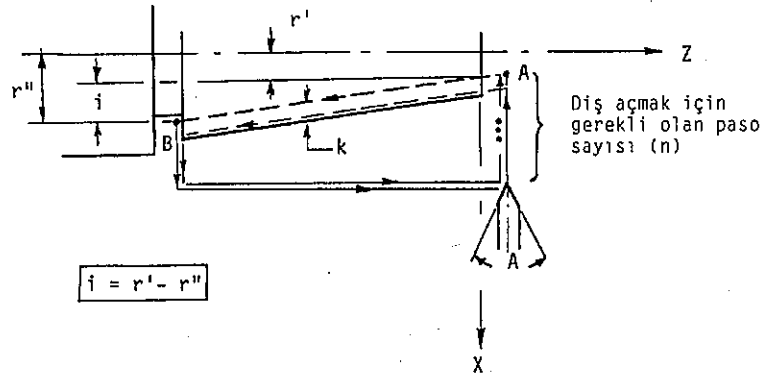
Şekil 2-34 Bir G73 kodu için I ve K kodlarının belirlenmesi.

G76 Kodu (Tekrarlamalı Diş Açma Çevrimi). Bu deyim formatı (Şekil 235):

$$G76XZdKkDd \left\{ \begin{array}{l} Ff \\ Ee \end{array} \right\} Aa^*$$

burada

- x ve z = vida sonunun koordinatları
- i = her iki uçtaki diş dibi panel
- k = diş yüksekliği (yarıçap değeri)
- d = ilk paso derinliği



Şekil 2-35 Bir G76 koduyla tanımlanan çoklu diş açma çevrimi.

- f ve e = diş adımı
- a = diş açısı

değerlerini ifade etmektedir.

Bu deyim belirtildiğinde; NC denetleyicisi, her bir paso için dişinin dış çapına ($x + 2k$) nazaran kesme derinliğini şu formüle göre otomatik olarak hesaplayacaktır.

$$\text{Kesme derinliği} = d \cdot (n)^{1/2}$$

burada; n sıra ile yapılacak paso sayısıdır. Buna göre, eğer ilk kesme pasosunun derinliği 0.5 mm ise, meteakip kesme derinlikleri:

$$\text{ikinci paso vermede: } (1/2) \cdot 2^{1/2} = 0.7071$$

$$\text{üçüncü paso vermede: } (1/2) \cdot 3^{1/2} = 0.8660$$

olacak ve diğerleri de bu formüle göre hesaplanacaktır. Diş yüksekliği 1.624 mm'dir. Gerekli paso sayısı:

$$n = (k/d)^2$$

formülüne göre belirlenir.

Bu formülde d ve k değerleri yerlerine konulduğunda:

$$n = (1.624/0.5)^2 = 10.549$$

elde edilir. Dolayısıyla, bu diş açma işlemi için onbir pasoya gerek vardır. Yukarıdaki hesap ve karar işlemleri NC denetleyicisi tarafından otomatik olarak yürütülür. Kısım 2.7.2'de belirtilen program yerine şu bir tek deyim kullanmak suretiyle, söz konusu diş açma işlemi gerçekleştirilmeye mümkündür:

G76X20.752Z-40.K1.624D0.5F3.A60*

FANUC 6T denetleyici için, kademeli delme (G74) ve kademeli kanal açma (G75) altyordamları da mevcuttur. Bu iki çevrimin bir torna tezgahında programlanması, bir freze tezgahı için yapılacak programlama işlemine benzer. Daha ayrıntılı bilgi için, okuyucunun kendi kullandığı NC tezgaha ait özel programlama kılavuzuna başvurması gerekir.

2.8 YARDIMCI FONKSİYONLARI TANIMLAYAN KODLAR

Kesici hareket yollarının tanımlanmasına ilave olarak; işlem modu, işleme özellikleri vb. gibi parametreleri tanımlamak için çok sayıda koda ihtiyaç duyulur. **M Kodu (Yardımcı Fonksiyon Kodu):** G kodu gibi, M kodu da çeşitli kullanımlara sahiptir. Genellikle M kodlarının anlamları, değişik imalatçıların NC tezgahlarına göre farklılık gösterir. Örneğin Tablo 26'da gösterilen seçilmiş M kodları, FANUC 6MB denetleyici sistemli Cadillac NC-300 freze tezgahı için ve FANUC 6T denetleyici sistemli Cadillac NC-100 torna tezgahı için kullanılır. M98 ve M99 kodlarının dışında, tabloda listelenmiş olan kodların her biri diğer kodlara gerek olmaksızın belirtilebilir. M98 kodu, bir ana programda bir altprogramı çağırarak için belirtilir. Söz konusu deyim formatı:

M98PpLl ';

burada *p*, bir altprogramın başlangıcında O kodu olarak belirtilen program numarası; *l* ise altprogramın kaç defa uygulanacağını gösteren sayıdır.

M99 kodu altprogramın son deyiminde belirtilir ve denetimin yeniden ana programdaki altprogramı çağıran deyimden hemen sonraki deyime geri dönmesini sağlar (Şekil 2-36). Denetimin ana programda altprogramı çağıran deyimden sonraki bir deyime yönlendirilmesi için, M99 kodundan sonra bir P kodu da belirtilebilir. Böylece, eğer altprogramdaki 12 etiketli deyimde tek bir M99 kodu belirtmek yerine,

M99P0032;

deyimi belirtilirse; denetim ana programda 0032 etiketli deyime gidecektir.

Altprogramlama, bir programı kısaltmak ve CNC denetleyicisinde bellek alanından tasarruf sağlamak için genellikle çok yararlıdır. Tipik örnekler aşağıda verilmiştir:

1. Bir parça profilinin bir freze tezgahında üç pasoda işlenmesi gerekmektedir. Freze kesicinin bu üç pasoda izleyeceği yollar X-Y düzleminde aynı olmakla beraber; kesme derinlikleri farklıdır. Bıçağın X-Y düzleminde yönlendirilmesi için bir altprogram yazılabilir. Kesicinin doğru Z konumuna her gelişinde, bu altprogram çağrılır.
2. Bir parça profilinin, aynı parmak freze kullanılarak bir freze tezgahında, biri kaba işleme ve diğeri de ince işleme olmak üzere iki pasoda işlenmesi gerekmektedir. Birisi kesici yarıçapı ile izin verilen işleme payının toplamına, diğeri de kesici yarıçapına eşit olan iki fark değeri, fark

Ana program	Alt program
.....	00008;
.....
N0030M98P0008L2;
N0031.....;
.....	N12M99;
.....

Şekil 2-36 Bir altprogramın kullanılması: 0008 nolu altprogram iki kez yürütülür ve sonra da denetim ana programdaki N0031 etiketli deyime geri döner.

yazmaçlarının belleğinde saklanmış durumdadır. Kesici yarıçapı farkı kodunu kullanarak kesici hareket yolunu tanımlamak için bir altprogram yazılabilir. İlkinde birinci fark değerini, sonra da ikincisini kullanarak aynı altprogramı iki kez yürürlüğe koymak suretiyle, kaba ve ince paso verme işlemleri gerçekleştirilebilir.

S Kodu. S kodu mil devrini denetler. Tezgahın özelliklerine bağlı olarak, S adresinden sonra iki basamaklı veya dört basamaklı bir sayı belirtilebilir. Devir sayısının birimi, takım tezgahı yapımına göre değişir. Örneğin Cadillac NC-100 torna tezgahında, devir sayısını en büyük devirin yüzde değeri cinsinden göstermek üzere iki basamaklı bir sayı kullanılır.

T Kodu. NC'li takım tezgahının tasarımına bağlı olarak, T adresinden sonra iki veya dört basamaklı bir sayı belirtilebilir. FANUC 6T denetim sistemli Cadillac NC-100 model torna tezgahında, dört basamaklı sayı belirtilebilmektedir. İlk iki basamak döner başlık konumunu veya kesici takım numarasını, son iki basamak ise fark numarasını gösterir.

2.9 NC PROGRAMLAMA ÖRNEKLERİ

NC programlama, normal olarak, tezgahta talaş kaldırma işlemlerinin NC kodları cinsinden adım adım yazılmasından ibarettir. Aslında bu, tezgahta işleme planının İngilizce'den NC kodlarına çevrilme işlemidir. NC programcılarının deneyim ve bilgi düzeyine bağlı olarak, aynı parça için farklı programcılar tarafından yazılan programlar birbirlerinden farklılık gösterebilir. Bir NC programının değerlendirilmesinde rol oynayan başlıca kriterler, doğruluk, hassasiyet ve programın CNC denetleyicisinde saklanması için ihtiyaç duyulan bellek alanıdır.

Bir NC programı, doğru biçimde bir kesici takım yolu üretmeli ve söz konusu parça için belirtilen boyut ve tolerans gereklerini karşılamalıdır. Çoğu CNC denetleyicilerinde sınırlı bir bellek kapasitesi (şerit boyuna dönüştürüldüğünde 80 m veya 160 m) mevcut olduğundan, programın olabildiğince kısa ve öz olmasını sağlamak önemlidir. En uygun koordinat sisteminin seçilmesi veya mevcut sistemin en uygun koordinat sistemine kaydırılması, altprogramların etkin biçimde kullanılması, gereksiz kodların (örneğin N kodu gereksizse çıkarılabilir) ve önceki deyimlerde tanımlanmış olan kodların çıkarılması, bir programın uzunluğunu önemli oranda azaltabilir.

Bir NC programı, örneğin nümerik kontrollü takım tezgahının karakteristik özellikleri ve yapısı, iş parçası tespit yöntemi, kullanılacak kesici takımlar, iş parçasının biçim ve malzemesi, ayar ve işletim kolaylığı, tezgahta işleme süresinin

kısaltılması ve emniyet gibi unsurları dikkate alan somut bir işleme süreç planı esasına dayalı olarak yazılmalıdır. Eğer tezgahta işleme süreç planı gereğine uygun değilse, NC programı da hiçbir zaman başarılı olamaz.

Bir nümerik kontrollü kesici takım yolunun elle programlanması sırasında NC programcısının yapması gereken başlıca iş, kesici takımın ardışık konumlarına ait koordinatları hesaplamaktır. G40 kodu programlanmış olduğu takdirde (yani kesici uç yarıçapı telafisi iptal edildiğinde); bir NC deyiminde belirtilen koordinatların, parmak freze için kesici uç yarıçapı merkezinin konumunu, torna kalem için ise uç yarıçap merkezinin konumunu belirlediğine dikkat edilmesi gerekir. Kesici uç yarıçap telafisi modunda programlama, kesicinin izlemesi gereken yolun programlanmasında parça profilinden yararlanılabilmesi mümkün olur. Diğer bir deyişle, NC deyimlerinde (bloklarında) belirtilmiş olan koordinatlar, parça profilinin koordinatlarıdır. Böylece, parça profilini esas alarak kesici takım konumlarını hesaplamak için gerekli olacak ek işten tasarruf sağlanabilir. Her iki durumda da bir NC programcısı, her bir NC deyim için varış noktasının koordinatlarını mutlaka hesaplamak zorundadır. Bu, özellikle parça profilinin karmaşık bir yapıda olması doğrusal ve dairesel biçimlerin dışındaki kesitlerden oluşması durumunda, ağır bir iştir. Genellikle bu hesaplama işi o kadar karmaşıktır ki, nümerik kontrollü kesici takım yolunu elle programlamak makul bir yöntem değildir. Bunun yerine; APT veya COMPACT II gibi daha yüksek düzeyli dillerde ya da bir bilgisayar grafik sistemini kullanmak suretiyle NC programlama yöntemi seçilmelidir.

TABLO 2-8 BOŞ BİR İŞLEM FÖYÜ

İşlem No	Takım No	Kesme Hızı	İlerleme hızı	Mil hızı	İşlemin tarihi	İş tutma

Bir NC programı yazılmadan önce; alışılmış uygulama; ardışık işleme adımlarını işlem föyü adı verilen bir tablo biçiminde (Tablo 2-8) tanımlamak için ihtiyaç

duyulan gerekli tüm bilgileri (kesici takım hareketi, kullanılan takımlar ve istenen işleme özellikleri dahil) bir liste halinde sıralamaktır. Özellikle bir takım tezgahında işleme programında birden çok kesici takım kullanılmakta ise, ayrı bir kesici takım föyüne de (Tablo 2-9) gerek vardır. Bu tabloda; kesici takım numaraları, ebatlar, ofset (kaçıklık) ve yarıçap telafisi değerleri, sert maden (karbür) uç tipleri, tutucular ve kaçıklık kaydedici numaraları (yani, ofset ve yarıçap kompanzasyon değerlerini bellekte saklamak için kullanılan yazmaçların adresleri) listelenir. Sonra da, işlem ve kesici takım föylerinde verilen bilgiye göre, elle yazılmış bir NC programı bir tablo biçiminde (örneğin Tablo 2-10'da gösterildiği gibi) yazılabilir.

TABLO 2-9 BOŞ BİR KESİCİ TAKIM FÖYÜ

Takım No.	Yan Takım No.	Anma Ölçüsü Takım Tanımı	Takım ofseti			Kesici Çapı ve Burun Yarı Çap Kompansasyonu	Giriş tipi	Tutucu tipi
			X	Y	Z			

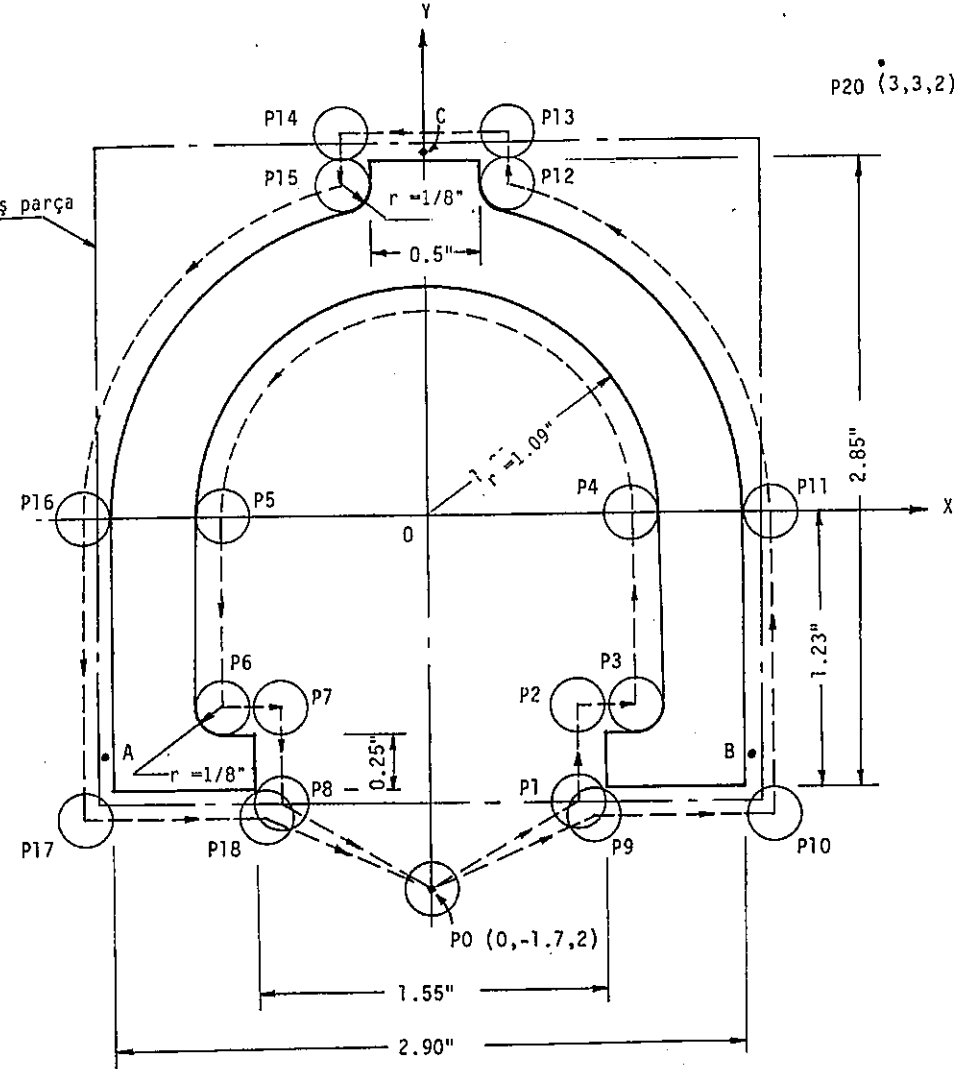
NC programlama detaylarını açıklamak için seçilmiş olan aşağıdaki örnekler²⁸, UCLA (University of California, Los Angeles)'da sayısal kontrol laboratuvar dersi için verilen alıştırmalardır. Birinci örnekte, kesici yarıçap telafisi modunda programlama açıklanmaktadır. İkinci örnek ise; bir delme işlemini programlamak için; programlama modunun kaydırılması (G90/G91), koordinat sisteminin değiştirilmesi, altprogramın ve tezgahta işleme çevrimi deyiminin yürürlüğe konulması dahil; çeşitli tekniklerin birlikte kullanımını göstermektedir.

TABLO 2-10 ELLE YAPILAN BİR NC PROGRAMLAMA FÖYÜNE AİT BİR ÖRNEK

O	N	G	X	Y	Z	F	S	T	M	P	Öneri
0001	10	90;92	0	-1.7000	2.0000		25		03			Program numarası
					0.1000				08			Mutlak programlama; mil hızını ve başlangıç ayarları: mili döndür.
												Hızla aşağı hareket et; soğutucuyu çalıştır.

Örnek 1

V-blok mengene iki adımda işlenmekte olup; birinci adımda, ham blok malzeme A, B ve C noktalarındaki tespit üniteleri tarafından sabitlenmekte ve parçanın iç profili, sırasıyla değişik talas derinliklerinde üç kademeli olarak işlenmektedir. Buna göre,



Şekil 2-37 Örnek 1 (Kısım 2.9)'deki 0002 ve 0004 No.lu programlar ile ilgili bir V-blok mengene parçası ve kesici yolları. Parçanın kalınlığı: 1/2 inç; parmak freze kesicisi: 1/4 inç; malzeme: Alüminyum levha, 3 x 3 x 1/2 inç.

O0001; (Program No. 0001)
 N10G90G92X0Y-1.7Z2.0M03S25; (mutlak koordinat programlama; kesici takımın başlangıç konumu: P0; tezgah koordinat sisteminin başlangıç noktasını O'a ayarla, devir sayısını maksimum hızın % 25'i olarak ayarla; iş milini saat yönünde çalıştırmaya başla.)
 N15G00Z0.1M08; (kesici takımı hızla $z = 0.1$ inç düzeyine indir; soğutucu sıvıyı devreye sok.)
 N20G01Z-0.18F2.0; (kesici takımı 2 inç/dak. olarak programlanmış ilerleme hızında $z = -0.18$ inç düzeyine indir.)
 N30G10P01R0.125; (1 nolu fark kaydedicisi içinde değerini, 0,125 inç'lik yarıçapı olarak ayarla.)
 N40M98P0002; (kesici takımı P0-P1-P2-...-P8-P0 hareket yolu boyunca yönlendiren alt program No. 0002'yi çağır.)
 N50G01Z-0.36; (kesici takımı daha da aşağı indirerek $z = -0.36$ inç düzeyine getir.)
 N60M98P0002; (altprogram No. 0002'yi çağır ve kesme işlemini tekrarla.)
 N70G01Z-0.52; (kesiciyi istenen paso derinliği olan $z = -0.52$ inç konumuna hareket ettir.)
 N80M98P0002; (altprogram No. 0002'yi çağır ve kesme işlemini tekrarla.)
 N90G00Z2.0M09; (kesiciyi ilk düzeyine gelene kadar yukarı kaldır ve soğutucu pompayı durdur.)
 N100X3.0Y3.0M05; (kesiciyi P20 noktasına getir.)
 N110M30; (program ve bellek geri sarımının sonu.)
 O0002; (Program No. 0002)
 N10G17G41D01G01X0.775Y-1.3; (X-Y düzlemini kesici yarıçapı telafi düzlemi olarak ayarla; profilin sol tarafındaki kesici yarıçap telafisini ayarla; 1 no.lu fark kaydedicisinde saklanmış olan değeri yarıçap telafisi için gerekli olan fark değeri olarak ayarla; kesici takımın P0'dan P1'e hareket etmesi için komut ver.)
 N20Y-0.98; (P1'den P2'ye hareket)
 N30X1.09; (P2'den P3'e hareket)
 N40Y0; (P3'den P4'e hareket)
 N50G03X-1.09R1.09; (P4'den P5'e dairesel enterpolasyon hareketi)
 N60G01Y-0.98; (P5'den P6'ya hareket)
 N70X-0.775; (P6'dan P7'ye hareket)
 N80Y-1.3; (P7'den P8'e hareket)
 N90G40G01X0Y-1.7; (P8'den P0'a hareket; kesici yarıçap telafisini bu hareketin sonunda iptal edilir.)
 N100M99; (denetimi, ana program içinde bu altprogramı çağırın deyimden hemen sonra yeralan deyim geri döndür.)

Parçanın orta kısmında bırakılmış olan blok malzemesi son pasoda P8 noktasından P0 noktasına kadar olan yolda kesici takım hareketini engelleyeceği için, malzemenin merkezi

kısımında talaş kaldırma işleminin yapılabilmesi amacıyla ilave bir kaba işleme pasosu vermek daha iyidir.

Parça iç profilden tezgaha tespit edildiği takdirde ve 0001 No.lu programda belirtilen başlangıç noktası ve koordinat sisteminin aynen seçilmiş olması halinde, dış profil de üç pasoda işlenebilir. Şekil 237'de, 0004 No.lu program tarafından tanımlanmış olan P0-P9-P10-...-P18P0 biçimdeki kesici takım hareket yolu da gösterilmektedir. 0003 No.lu program, yukarıda verilen 0001 No.lu programa benzer. Bu programlar şöyledir:

O0003;
 N10G90G92X0Y-1.7Z2.0M03S25;
 N15G00Z0.1;
 N20G01Z-0.18F2.0;
 N30G10P01R0.125;
 N40M98P0004;
 N50G01Z-0.36;
 N60M98P0004;
 N70G01Z-0.52;
 N80M98P0004;
 N90G00Z2.0;
 N100Y2.0M05;
 N110M30;
 O0004;
 N10G17G41D01X0.75Y-1.23;
 N20X1.45;
 N30Y0;
 N40G03X0.25Y1.4283R1.45;
 N50G01Y1.62;
 N60X-0.25;
 N70Y1.4283;
 N80G03X-1.45Y0R1.45;
 N90G01Y-1.23;
 N100X-0.75;
 N110G40G01X0Y-1.7;
 N120M99;

Örnek 2

Şekil 2-38(a)'da gösterilen parça özel bir püskürtme memesi (nozül) için dizayn edilmiş olup; her biri 0.01 inç çapında 500 deliği vardır. Bu delikler aynı diziye sahip beş gruba ayrılmıştır. (0002 No.lu) altprogram tarafından tanımlanan kesici takım hareketi Şekil 2-38(b)'de gösterilmiştir. Bu programı tekrarlı olarak 10 kez yürütmek suretiyle bir gruptaki 100 deliğin, tamamı delinebilir. Altprogramın 10 kez yürütülmesinden sonra, kesici takım hareketi bitiş noktasının P7 noktası olduğuna dikkat ediniz. Herbir grupta

yer alan deliklerin delinmesi için, tezgahın koordinat sistemi söz konusu grubun merkezine ayarlanır. Dolayısıyla, koordinat sisteminin bu programda yeniden tanımlanması gereklidir. Bu 500 deliğin delinmesi için kullanılacak NC programları şöyledir:

O0001; (ana program)

G90G92X0Y0Z0.5M03S50; (başlangıç noktası (0,0,0.5) olan mutlak koordinat sisteminde programlama; tezgah koordinat sisteminin başlangıç noktasını 0'a ayarla; devir sayısının maksimum hızın % 50'sine ayarla; iş milini saat yönünde çalıştır)

G00X0.0354Y-0.3536Z0.05M08; (kesici takımı [0.0354,-0.3536,0.05] noktasına hareket ettir; soğutucu sıvıyı devreye al)

M98P0002L10; (alt programı 10 kez yürüt ve alt diziyi del)

G90G92X0.3889Y-1.0Z0.05; (mutlak koordinatlarda programlama yaparak, tezgah koordinat sisteminin başlangıç noktasını P12 noktasında sıfırla)

G00X0.0354Y-0.3536; (kesici takımı [0.0354,-0.3536,0.05] noktasına hareket ettir)

M98P0002L10; (alt programı 10 kez yürüt ve üst diziyi del)

Şekil 2-38 (a) Aynı dizide beş gruba bölünmüş 500 delikli bir meme plakası. Kalınlık 0.15 inç; Z yüzey = 0. (b) Altprogram tarafından tanımlanan kesici takım hareketi. P1, P4, P6 ve P7 noktalarının delik konumları olmadığına dikkat ediniz.

G90G92X0.3889Y2.0Z0.05; (mutlak koordinatlarda programlama yaparak; P14 noktasında tezgah koordinat sisteminin başlangıç noktasını sıfırla)

G00X0.0354Y-0.3536; (kesici takımı [0.0354,-0.3536,0.05] noktasına hareket ettir)

M98P0002L10; (alt programı 10 kez yürüt ve alt diziyi del)

G90G92X1.3889Y-1.0Z0.05; (mutlak koordinatlarda programlama yaparak; P13 noktasında tezgah koordinat sisteminin başlangıç noktasını sıfırla)

G00X0.0354Y-0.3536; (kesici takımı [0,0354,-0.3536,0.05] noktasına hareket ettir)

M98P0002L10; (alt programı 10 kez yürüt ve sol diziyi del)

G90G92X-1.6111Y0Z0.05; (mutlak koordinatlarda programlama yaparak, P11 noktasında koordinat sisteminin başlangıç noktasını sıfırla)

G00X0.0354Y-0.3536; (kesici takımı [0,0354,-0.3536,0.05] noktasına hareket ettir)

M98P0002L10; (alt programı 10 kez yürütün ve sağ diziyi del)

G90G92X1.3889Y0Z0.05M09; (mutlak koordinatlarda programlama yaparak; 0 noktasında tezgah koordinat sisteminin başlangıç noktasını sıfırla; soğutucu sıvısını kapat.)

G00X0Z0.5M05; (başlangıç konumuna geri dönerek; milini durdur)

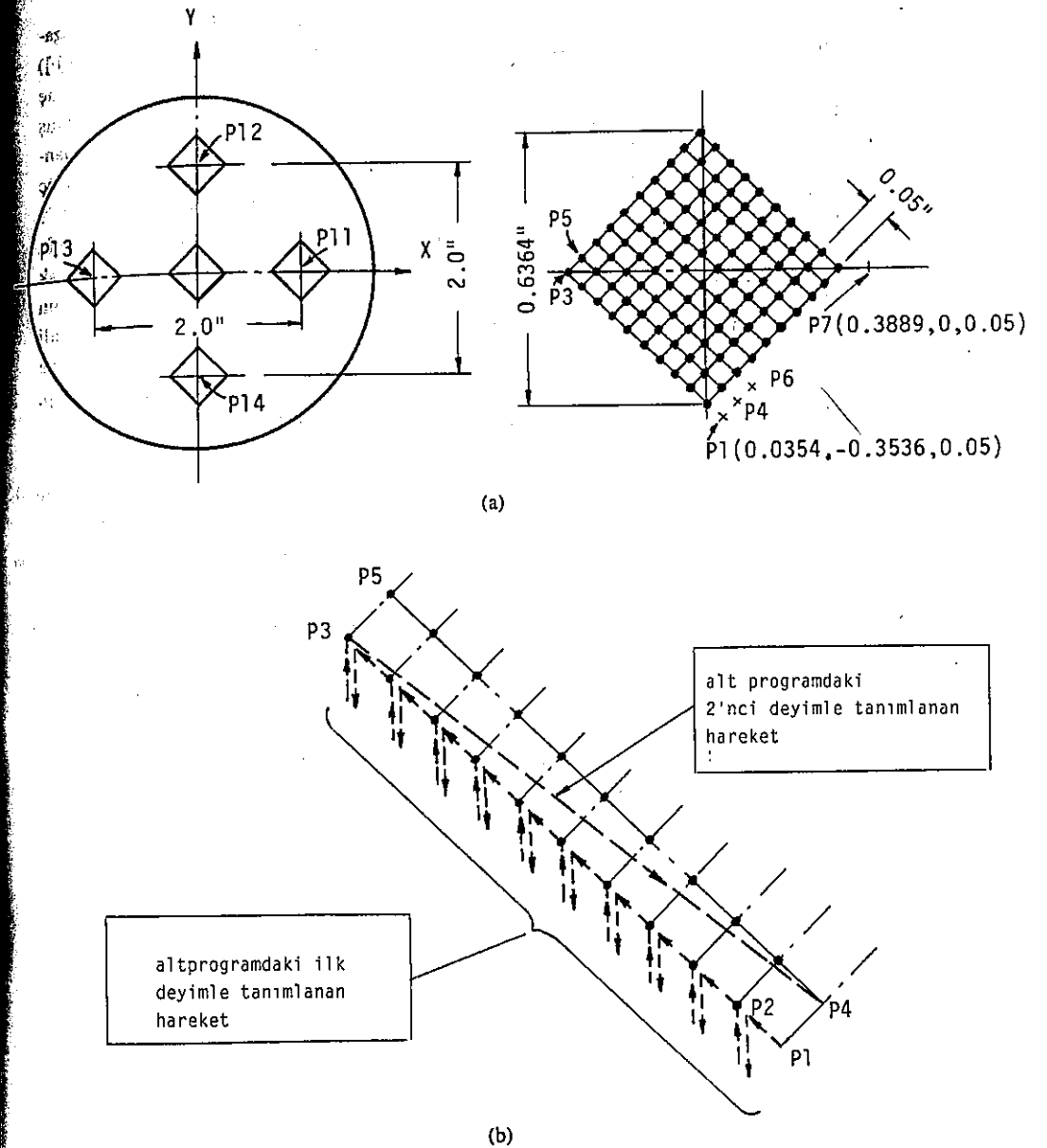
M30; (program ve bellek geri sarımının sonu)

O0002; (alt program)

G91G99G83X-0.0354Y0.0354Z-0.21Q0.02R0F1.0L10; (artışlı koordinatlarda programlayarak; -1 eğimli doğru (örneğin Şekil 2-38[b]'de gösterilen P2P3 doğrusu) üzerine kademeli olarak on delik del)

G00X0.3889Y-0.3182; (diğer bir delme çevrimine başlamak üzere hazır konuma dön)

M99; (çağırma programında bu programı çağırın deyimden hemen sonraki deyime dön)

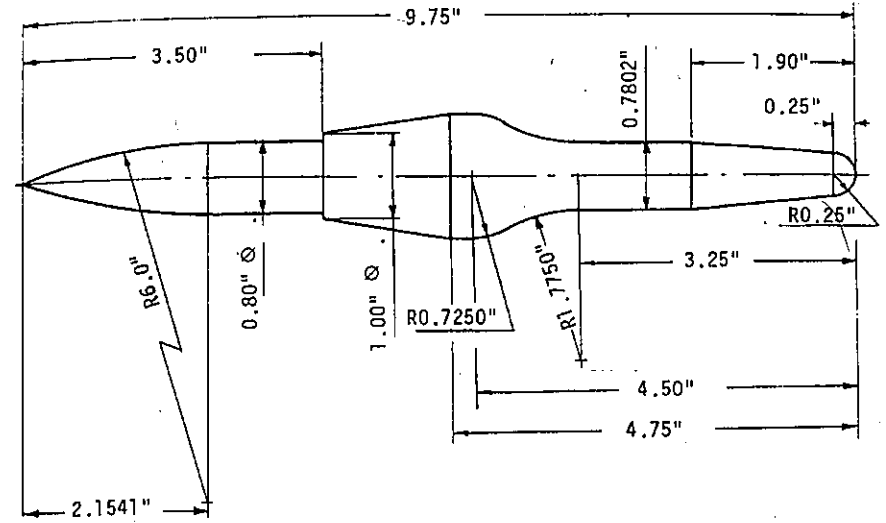


Şekil 2-38

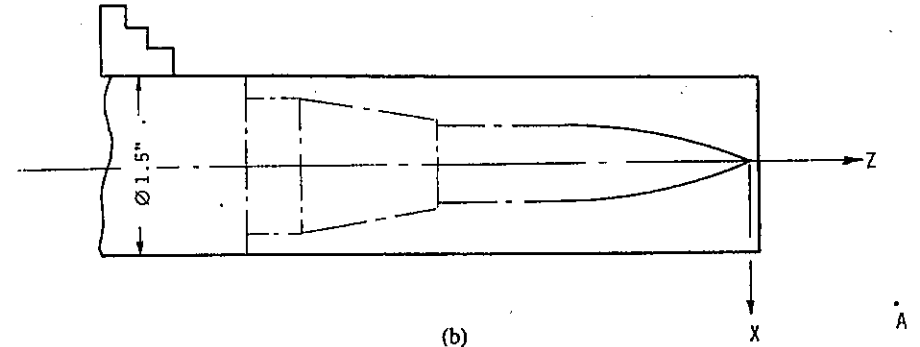
Örnek 3

Şekil 2-39(a)'da gösterilen parça, FANUC 6T denetleyicili NC-100 torna tezgahında işlenecektir. Bu parçanın işlenmesi için; sol tarafın tornalanması için (Şekil 2-39(b)) O0001, sağ tarafın tornalanması için de (Şekil 2-39(c)) O0002 nolu programa ihtiyaç vardır. Bu tornalama işleminde sırasıyla 1 ve 3 No.lu döner başlık konumlarında takılan olan işleme kalem kaba ve ince işleme kalem kullanılmaktadır. Herbir programın başlangıcında; kaba kesici takım çalışır durumda iken, koordinat sisteminin başlangıç noktası uç yüzeyin merkezine ayarlanmaktadır. Her iki programda da kalem uç yarıçapları sıfır varsayılmakta olup; böylece kalem yarıçap telafisine gerek yoktur. İnce işleme kaleminin numarası O3'dür ve iki kalem arasında X ve Z koordinatları bakımından ortaya çıkacak farklılıklar, telafi edebilmek için O1 numaralı bir fark değeri belirtilmiştir. Bu iki programın ilk deyiminde, kalemin referans noktasına geri dönmesi için komut verilmektedir. Çeşitli kodların kullanımı konusunda açıklayıcı bilgi verilmesi açısından, bu programda, kaba ve ince işleme kalemlerine ait hareket yollarını tanımlamak için birkaç kod kullanılmış olup; bu programlar, açıklamalarıyla birlikte aşağıda verilmektedir:

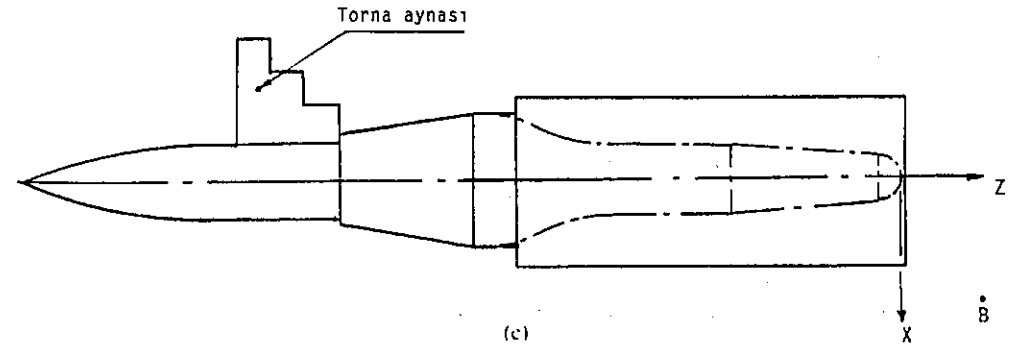
O0001* (Şekil 2-39(b)'de gösterildiği gibi sol tarafın tornalanmasına ait program)
 N1G20G28U0W0M42S40* (inç birimi cinsinden programlayarak; referans noktasına geri dön; devir sayısını maksimum hızın % 40 olarak ayarla)
 N2G50X15.8328Z12.6055* (koordinat sisteminin başlangıç noktasını uç yüzeyin merkezine ayarla)
 N3G00X1.6Z0.1M03* (kalemi, tornalama işlemine başlamaya hazır konuma getir; iş milini çalıştır)
 N4G94X0Z0F0.003* (uç yüzeyinde alın tornalamasını yap)
 N5G90X1.6Z-3.0I-0.1* (G90 kodu vasıtasıyla talaş kaldırma işlemi başlar)
 N6Z-3.5I-0.2*
 N7I-0.3*
 N8Z-3.75I-0.4*
 N9Z-4.0I-0.5* (talaş kaldırma sona erer)
 N10G00Z1.0* (G90 kodunu iptal et ve kalemi, $x = 0.8$ ve $z = 1.0$ koordinatlarıyla tanımlanan konuma getir)
 N11G73P12Q18I0.20K0U0.02W0.01D5F0.003S40* (diziyi tekrarlamaya başla; radyal geri çekme -- 0.2 inç; aksel geri çekme -- 0; radyal doğrultuda izin verilen son işleme payı -- 0.01 inç; aksel doğrultuda izin verilen son işleme payı -- 0.01 inç; kalem hareket yolu N12'den N18'e kadar etiketli deyimlerde tanımlanmıştır)



(a)



(b)



(c)

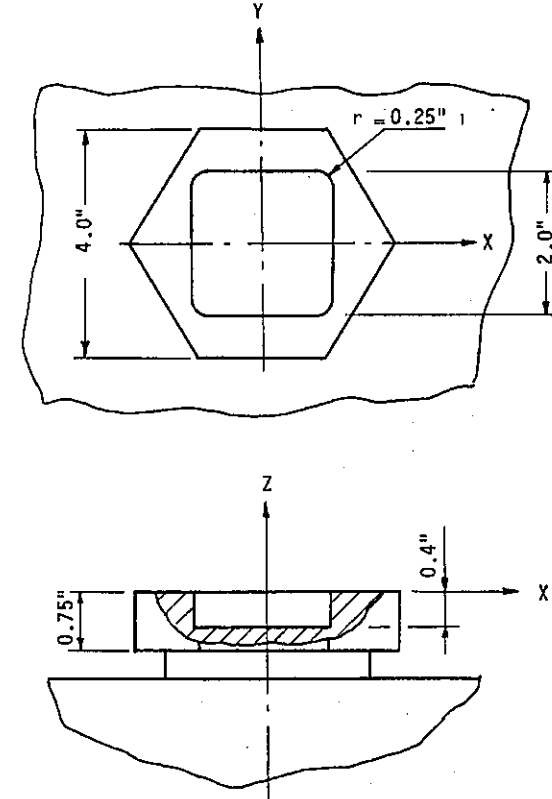
Şekil 2-39 (a) Tornalanarak biçimlendirilmiş mil. (b) Şekil 2-39(a)'da gösterilen parçanın sol baş işlemleri için kurna diyagramı. A noktası: (15.8328, 12.6055). (c) (a)'da gösterilen parçanın sağ baş kesimi için kurna diyagramı. B noktası: (15.8328, 12.3641).

N12G00X0Z0.05S50*
 N13G01Z0*
 N14G03X0.8Z-2.1541I-5.6K-2.1541*
 N15G01Z-3.5*
 N16X1.0*
 N17X1.45Z-5.0*
 N18Z-5.5* (dizi tekrarlamanın sonu)
 N19M11* (döner başlığı 90 derece çevir)
 N20M11T0301* (döner başlığı 90 derece daha çevir; son işleme kalemi çalışma konumuna girer; No. 1 fark değeri kullanılır)
 N12G70P12Q18F0.002* (ince işleme)
 N22G00X3.0Z3.0M05* (kalemi emniyetli konuma geri çek; iş milini durdur)
 N23M11*
 N24M11* (döner başlığı geri döndürerek tekrar başlangıç konumuna getir)
 N25M30* (program durur)
 O0002* (Şekil 2-39[c]'de gösterildiği gibi, sağ tarafın tornalanmasına ait program)
 N1G20G28U0W0* (tekrar B referans noktasına dön)
 N2G50X15.8328Z12.3641M42S40* (başlangıç noktasını ve devir sayısını ayarla)
 N3G00X1.6Z0.1M03*
 N4G94X0Z0F0.003* (alın işleme çevrimi)
 N5G71P6Q9U0D700F0.015* (çok tekrarlı tornalama çevrimini başlat)
 N6G00X0.5*
 N7G01X0.9Z-1.9*
 N8Z-3.25*
 N9X1.4Z-4.1375* (çok tekrarlı tornalama çevriminin sonu)
 N10G73P11Q17I0.075K0U0.02W0.01D2F0.003S40* (diziyi tekrarlama başla)
 N11G00X0Z0.1S50*
 N12G01Z0*
 N13G03X0.5Z-0.25I0K-0.25*
 N14G01X0.7802Z-1.9S45*
 N15Z-3.25*
 N16G02X1.2557Z-4.1375I1.7750*
 N17G03X1.45Z-4.5I-0.6279K-0.3625S40* (dizi tekrarlamanın sonu)
 N18M11*
 N19M11T0301*
 N20G70P11Q17F0.002* (son kesim pasosu)
 N21G00X3.0Z3.0M05*
 N22M11*
 N23M11*
 N24M30*

PROBLEMLER

2.1 Bir makine elemanının altıgen dış profilli ve kare biçiminde iç profilli olan parçası Şekil P2-1'de gösterilmektedir. Bu parçanın üst yüzeyinin istenilen ölçülere işlenmiş olduğunu varsayınız. Aşağıdaki parametreler öngörülmektedir:

Kesici takım	1/2 inç parmak freze
Devir sayısı	1000 dev./dak. (veya S34 no.lu NC kodu)
Kesici ilerleme miktarı	4 inç/dak. (kaba kesim) 2.5 inç/dak. (son kesim)
	2.0 inç/dak. (daldırma)
Maksimum talaş derinliği	0.4 inç



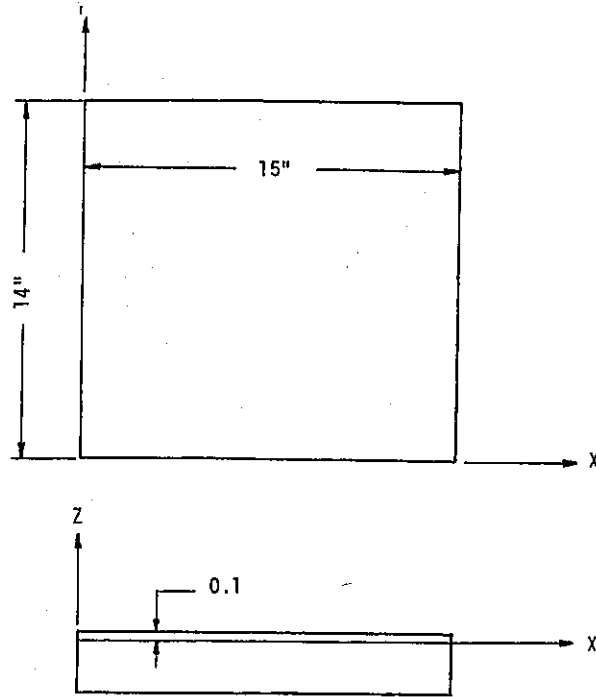
Şekil P2-1

Tezgahta otomatik kesici değiştirme tertibatı yoktur. Dış profilin (kaba işleme için iki paso, ince işleme için bir paso olmak üzere) toplam üç pasoda işlenmesi için bir NC programı yazınız. Kesicinin başlangıç konumu (0,0,2.0) noktasıdır.

- 2.2 Şekil P2-2'de görülen parça, 2 1/2 inç çapındaki bir parmak freze ile işlenecektir. Aşağıda verilen parametreleri kullanarak, bu yüzeyin işlenmesi için mümkün olan en kısa ve öz bir NC programı yazınız:

Devir sayısı	900 dev./dak. (veya S30 no.lu NC kodu)
İlerleme hızı	4 inç/dak.
Başlangıç konumu	(-5.0,0,3.0)
Talaş derinliği	0.1 inç

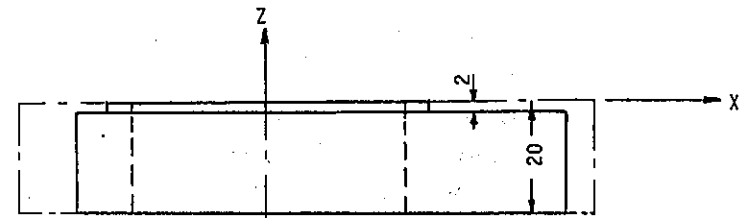
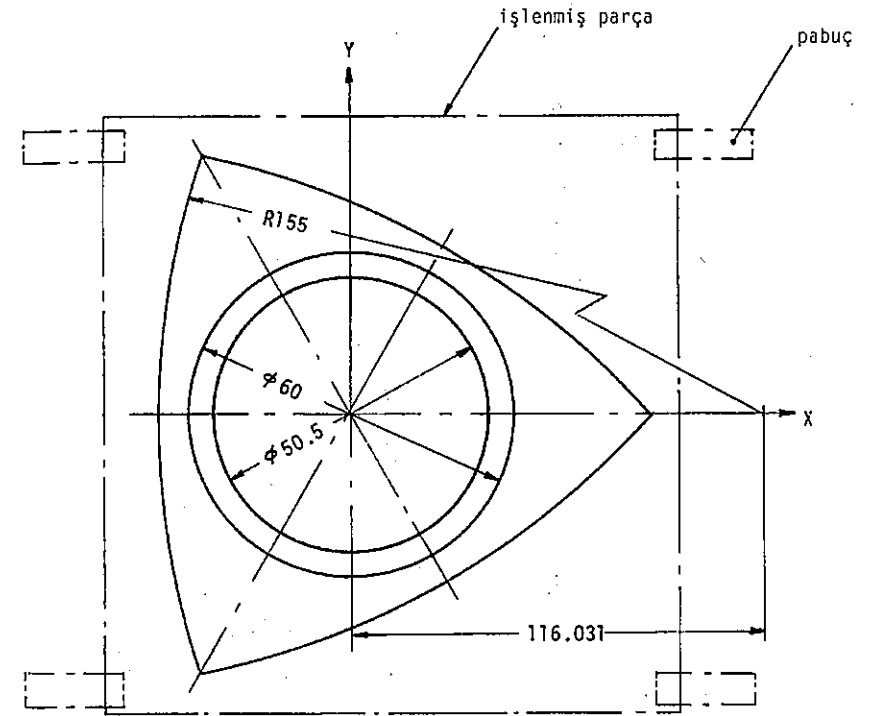
Yüzey bir pasoda işlenecektir.



Şekil P2-2

- 2.3 Bir Wankel (rotatif) motor rotorunun profili Şekil P2-3'de gösterilmiştir. Bu parçanın tezgahta işlenmesi için; biri üst yüzeylerin ve deliđin işlenmesi ve diđeri de dış profilin işlenmesini sağlamak üzere iki program ihtiyacı duyulmaktadır. Aşağıda belirtilen işleme şartları dikkate alınacaktır:

- Başlangıç ve bitiş noktaları (0,0,50.0) koordinat değerlerine sahiptir.
- Parçanın taban yüzeyinden itibaren 20.0 mm ve 18.0 mm ölçülerindeki üst yüzeylerini işlemek ve 50.05 mm çapında deliđin delmek için, 3/4 inç çapında bir parmak freze kullanılacaktır. Deliđin işlenmesi, iki kaba ve iki de ince işleme olmak üzere dört pasoda yapılacaktır. Yüzeyler ise sadece bir tek pasoda işlenecektir.
- Aynı freze, dış profili, (ikisi kaba ve ikisi de ince işleme olmak üzere) toplam dört pasoda işlemek için de kullanılmakta olup; herbir paso için işleme derinliđi 10 mm'dir. Parça, merkez deliđinin içinden



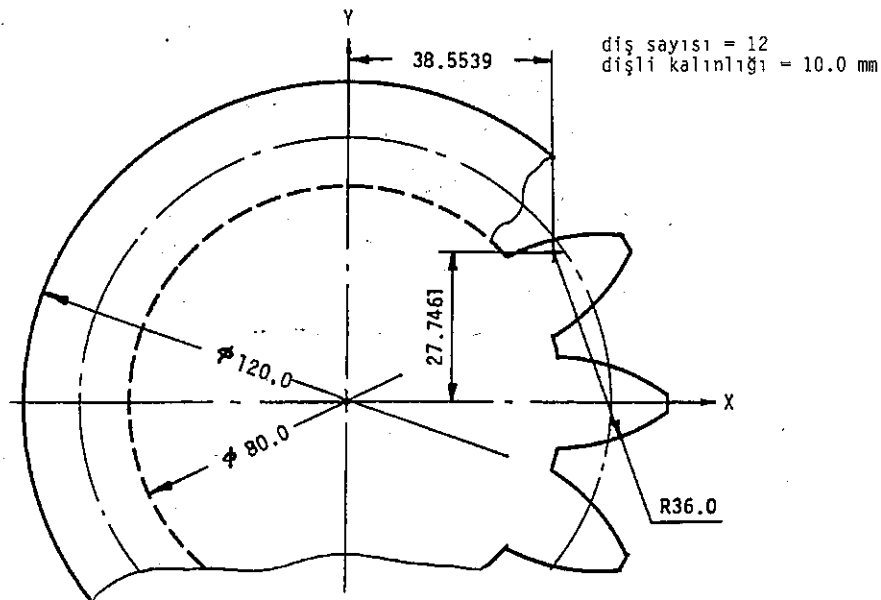
Şekil P2-3

doğru tezgaha tespit edilecektir. Sırasıyla kaba ve ince işlemler için, aynı ve zıt yönlü frezeleme yöntemleri kullanılmaktadır.

- d. Kaba ve ince işlemler için kesici ilerleme hızları, sırasıyla 100 mm/dak. ve 70 mm/dak.; daldırma için kesici ilerleme hızı ise 40 mm/dak.'dır. Devir sayısı 600 dev./dak. dır (veya S20 no.lu NC kodu).

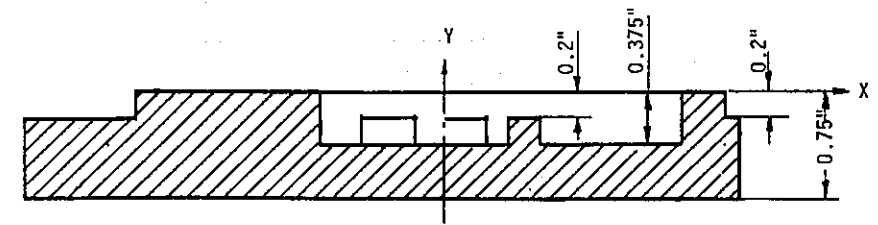
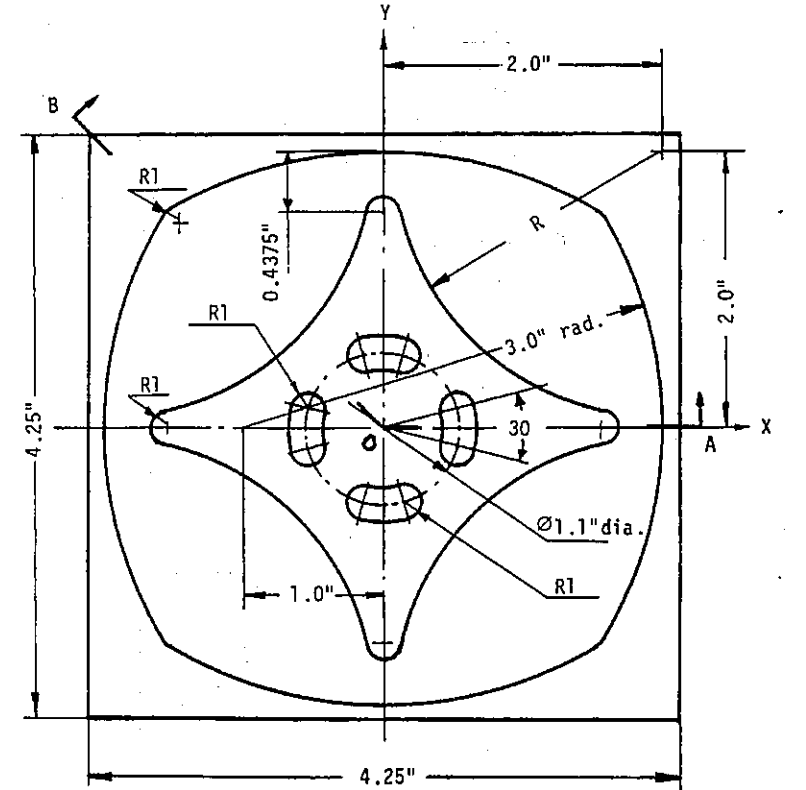
- 2.4 Basit diş profilini bir dişli Şekil P2-4'de gösterilmektedir. Dişli profili, 1/8 inç çapında bir parmak freze kullanılarak, (yedisi 1.5 mm'ye eşit kesme derinliğinde kaba kesim için, dördü de 3.0 mm kesme derinliğinde son kesim için olmak üzere) onbir pasoda imal edilecektir. Aşağıdaki parametreleri kullanarak suretiyle bir NC programı yazınız.

Devir sayısı	1500 dev./dak. (veya S50 no.lu NC kodu)
İlerleme hızı	40 mm/dak.
İnce talaş için izin verilen işleme payı	0.062 mm
Başlangıç noktası	(0,0,50.)
Üst yüzeyin Z koordinatı	0



Şekil P2-4

- 2.5 Şekil P2-5'de görülen bir matris, otomatik kesici takım değiştirici olmayan üç eksenli bir NC'li freze tezgahında işlenecektir. Üst yüzeyden, 0.2 inç mesafede ve (3 inç yarıçaplı) dairesel profilde olan yüzey, 3/4 inç çapında bir parmak freze vasıtasıyla bir pasoda işlenecektir. Geriye kalan yüzeylerin üç pasoda frezelenmesi için 1/4 inç çapında bir parmak freze kullanılmaktadır. Ham blok malzemenin istenilen 4.25 x 4.25 x 0.75 inç ölçülerinde işlenmiş ve tezgah mengeneyle bağlanmış olduğunu varsay-



A - O - B KESİT GÖRÜNÜŞÜ

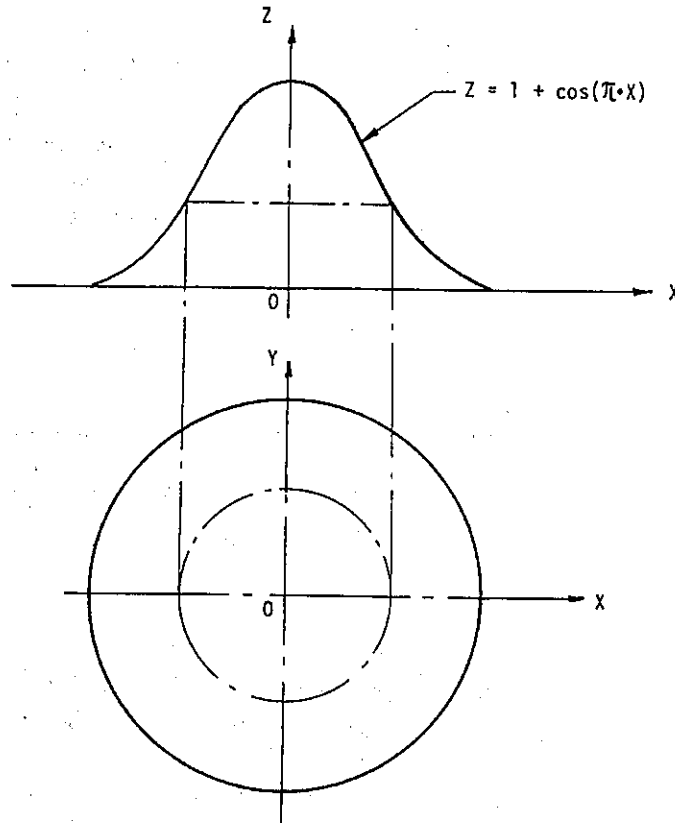
Şekil P2-5

yınız. Başlangıç noktası (0,0,1.0)'dir. Aşağıdaki parametreleri kullanarak, bu matrisin frezede işlenmesini sağlayacak NC programlarını yazınız.

KESİCİ ÇAPİ	DEVİR SAYISI	İLERLEME HIZI
$\frac{1}{4}$ "	900 dev./dak. (veya S30)	2.0 inç/dak.
$\frac{3}{4}$ "	300 dev./dak. (veya S10)	2.0 inç/dak.

R1 yarıçapı = 0.125"

2.6 P2-6'da görülen çan şekilli bir parça, 1/2 inç çapında bir yuvarlak uçlu kesicisi olan üç eksenli bir NC'li freze tezgahında işlenecektir. Problemin basitleştirilmesi bakımından, freze kesicisini şekilde gösterilen yatay kesitler (çemberler) etrafında, iki kesit arasındaki artışı mesafe 0.2 inç değerine eşit olmak üzere hareket ettirmek suretiyle parça profili



Şekil P2-6

yaklaşık değere getirilecektir. Aşağıdaki parametreleri kullanarak, son işlemeyi yönlendirecek bir NC programı yazınız.

Devir sayısı 1000 dev./dak. (veya S34 no.lu NC kodu)
İlerleme hızı 2 inç/dak.
Başlangıç noktasının (0,0,5.0)'da olduğunu kabul ediniz.

2.7

Bir Malta fasıllı hareket mekanizmasının hareket çarkı, Şekil P27'de gösterilen profile sahiptir. Parça önceki bir işlemde istenilen ölçülere göre işlenmiş bir merkezi delik içinden geçen bir bağlama aparatı ile yerine tespit edilmektedir. Gerekli frezeleme işlemi yönlendirecek bir NC programı yazınız. Aşağıdaki şartlar verilmiştir:

Kesici (T01)	3/4 inç'lik düz uçlu freze (deliklerin ve dış profilin işlenmesi için)
Kesici (T02)	1/4 inç'lik düz uçlu freze (yarıkları açmak için)
Kesici ilerleme hızı	100 mm/dak. (T01 kesicisi ile kaba frezeleme için) 70 mm/dak. (T01 kesicisi ile ince işleme için) 50 mm/dak. (T02 kesicisi için)
Dış profile izin verilen ince işleme payı	0.2 mm
Başlangıç ve bitiş noktaları	(150.0,150.0,30.0)

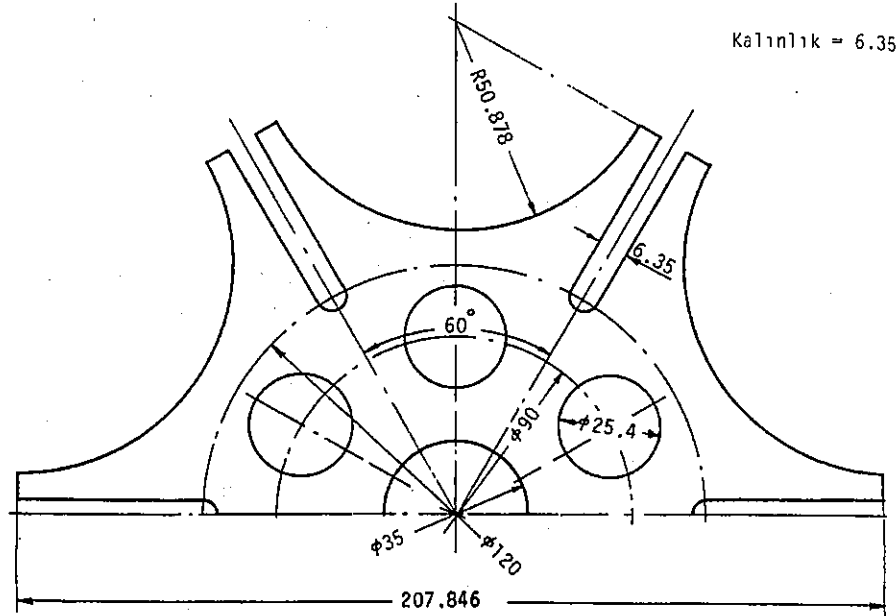
T02 no.lu kesici T01 no.lu kesiciden daha uzun olup; aradaki fark 30.988 mm'dir. Koordinat sisteminin başlangıç noktası, üst yüzeyin merkezidir.

Not: Bir FANUC 6MB denetleyicili NC freze tezgahında, iki bıçağın boyları arasındaki Δz fark (ofset) değerine ait telafi,

G43Hnn

kodları kullanılarak programlanabilir. Burada; H kodundaki "nn" sayısı, kesici boyları arasındaki Δz fark (kayma) değerinin bellekte saklanması sağlayan nn numaralı kaydediciyi temsil eder. Bu deyim etkisi, müteakip deyimler tarafından tanımlanmış kesici takım konumlarının z koordinatlarına Δz değerine ilave etmektedir. Kesici takım boyuna ait telafiyi sağlayabilmek için, Hnn kodu ile birlikte bir G44 kodu da kullanılabilir. Ancak, bu kodun etkisi, müteakip NC deyimleriyle

tanımlanmış olan z koordinatlarından Δz fark değerini çıkarmaktır. Kesici takım boyuna ait telafiyi iptal etmek için, bir G49 veya H00 kodu kullanılabilir.



Şekil P2-7

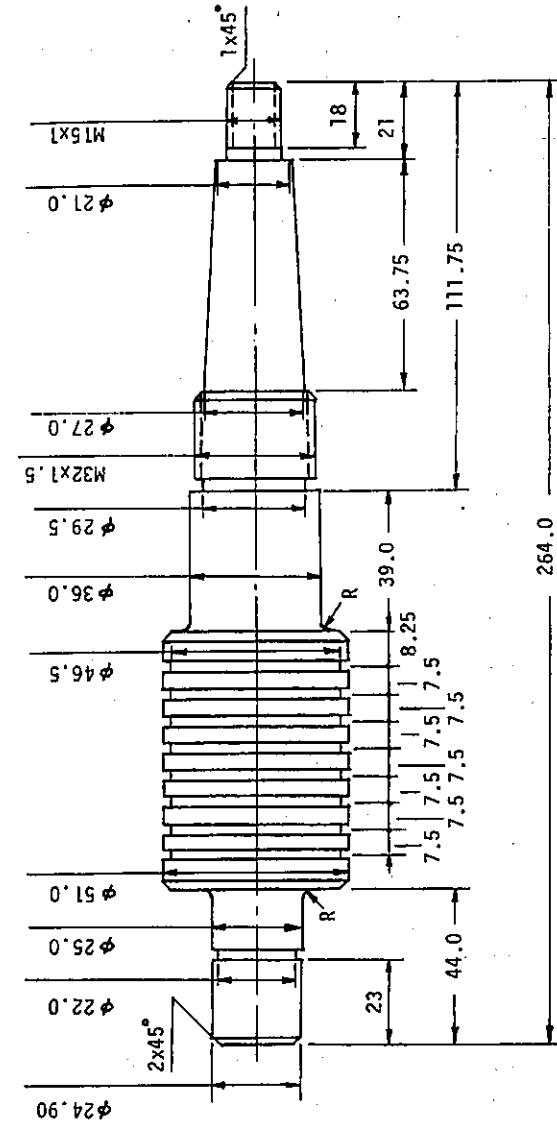
2.8 Şekil P2-8'de gösterilen mil, bir NC'li torna tezgahına merkezlerinden puntalamak suretiyle bağlanacak ve tezgahta işlenecektir. Sırasıyla kaba tomalama, ince işleme, kanal açma ve diş açma işlemleri için T01, T02, T03 ve T04 no.lu dört kalem kullanılmaktadır. Koordinat sistemi kaba işleme kalemi esas alınarak ayarlanmış olup; bu dört kaleme ait X ve Z fark değerleri, kalem yarıçapları ve kesici vektör no.ları aşağıdaki tabloda gösterildiği gibidir.

KAYDEDİCİ	X FARKI (mm)	Z FARKI (mm)	YARIÇAP (mm)	KESİCİ VEKTÖR NO.
1	0	0	0.25	2
2	+3.304	+3.995	0.13	2
3	-5.110	+13.573	0	9
4	-4.912	+12.517	0	9

Ham blok malzeme 55 mm çapında olup; parça iki NC programı kulla-

belirtilmemiş havşa
kanalların genişliği R

1.5x45
2.5
2.0



Şekil P2-8

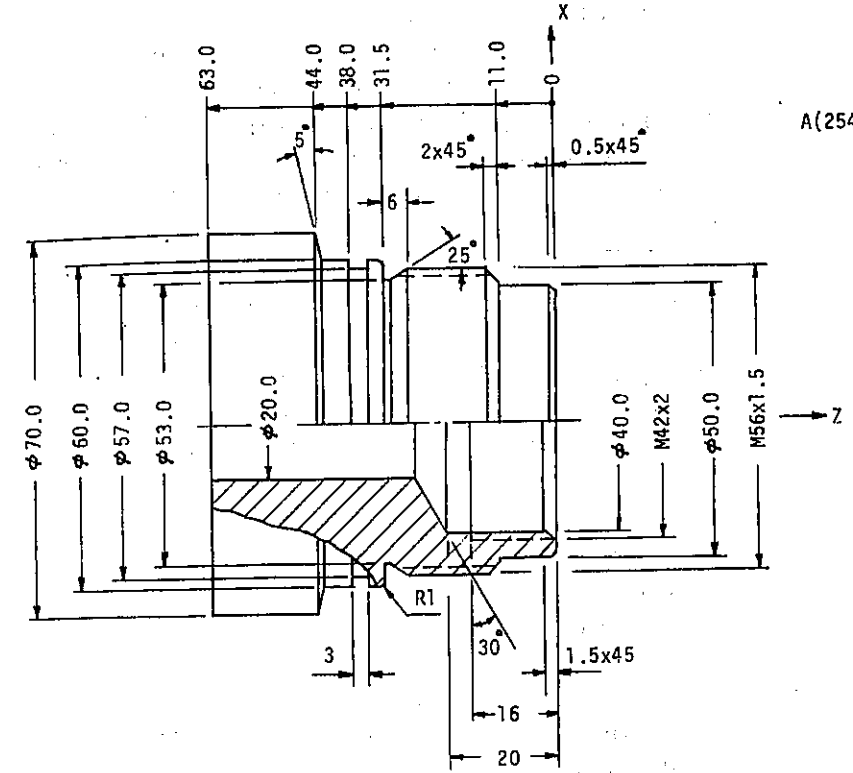
nılarak işlenecektir. İlk program vasıtasıyla, parça Şekilde gösterildiği gibi konumlandırılmakta ve $z = 0$ 'dan $z = -220$ mm'ye kadar olan profil tormalanmaktadır. Sonra da mil, şekilde gösterilen sol uç punta tarafından desteklenecek biçimde ters çevrilmekte ve ikinci bir program, $z = 0$ (uç yüzey)'den $z = -45.5$ mm'ye kadar olan parça profilini tormalamak için kullanılmaktadır. Bu iki programa ait koordinat sistemlerinin başlangıç noktası, sağ uç yüzeyin merkezine ayarlanmaktadır. İki punta deliğinin delinmiş olduğunu ve aşağıdaki parametrelerin kullanıldığını varsayınız.

İlerleme hızı	0.13 mm/dev. (kaba tormalama)
	0.06 mm/dev. (ince işleme)
	0.02 mm/dev. (kanal açma)
En büyük talaş derinliği	3 mm (kaba işleme)
Devir sayısı	800 dev./dak. veya S40 (kaba işleme)
	1200 dev./dak. veya S60 (ince işleme)
	400 dev./dak. veya S20 (kanal açma ve diş açma)
İzin verilen son işleme payı	0.2 mm
Referans noktasının koordinatları	(201.076,191.8)

2.9

Şekil P2-9'da verilen parça, gerek iç ve gerekse bir dış profile sahiptir. Bu parçayı üretmek için, FANUC 6T denetleyicili bir NC tormalama merkezi kullanılmaktadır. Tormalama işleminde kullanılan kesici takımlar ve bunların fark kayıtları şöyledir:

T0101	Dış yüzey kaba tormalama ve alın işleme
T0202	Delik delme (çap 20.0 mm)
T0303	İç yüzey kaba tormalama
T0404	İç yüzey ince işleme
T0505	İç yüzeye dış açma
T0606	Dış yüzey ince işleme ve 25 derece konik yüzeyli kanal açma
T0707	Dış yüzeye dış açma
T0808	Dış yüzey kanal açma ve kesme kalemni (genişlik = 3 mm)



Şekil P2-9

Ham blok malzeme 75 mm'lik bir çapa sahip olup, aynaya tespit edilmiştir. Parça, son işlemeden sonra ham bloktan kesilip alınmaktadır.

Aşağıdaki parametreleri kullanarak, bu parçanın işlenmesini sağlayacak bir NC programı yazınız:

İŞLEM	DEVİR SAYISI (NC KODU OLARAK)	KESİCİ İLERLEME HIZI (mm/dev.)
Kaba tormalama		
Dış yüzey	S700	0.13
İç yüzey	S900	0.10
İnce işleme	S1000	0.05
Kanal açma ve kesme	S300	0.04
Dış açma	S300	
Delik delme	S300	0.08

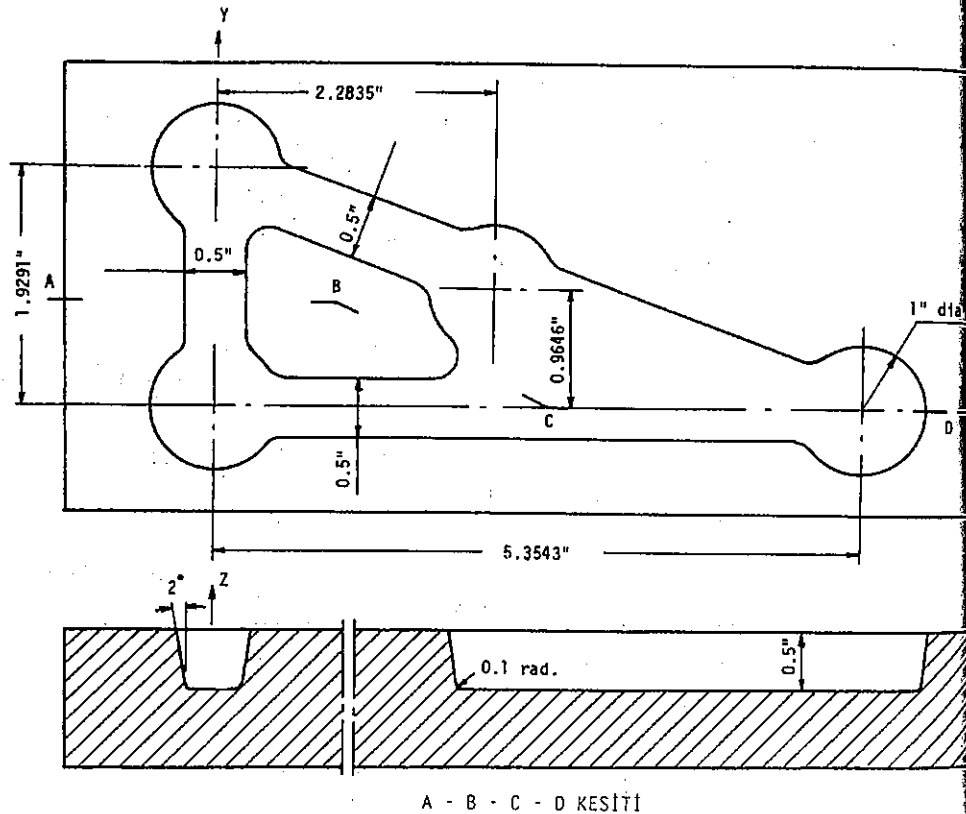
Takım değiştirme işlemi, şu iki durumda programlanmalıdır:

1. Referans noktası A'ya geri dönülmesi (G28_W_)
2. Takımın değiştirilmesi (M06T_)

2.10 Şekil P2-10'da gösterilen matrisin boşluk kısmı, üç eksenli bir NC'li freze tezgahında işlenerek yapılacaktır. Tezgah işlemlerini yönlendirecek bir NC'li programı yazınız. Aşağıdaki parametreler verilmiştir:

Kesici takım	0.1 inç köşe yarıçaplı, 1/2 inç'lik düz uçlu freze bıçağı
İlerleme hızı	2.5 inç/dak.
Devir sayısı	S30
Başlangıç noktası	(-5.0,5.0,3.0)

Yüzey üzerinde izin verilebilir sapma kenar yüksekliği 0.005 inç.



Şekil P2-10

KISIM - I İÇİN KAYNAKLAR

1. Modern Machine Shop 1987 NC/CIM Guidebook. Cincinnati, Ohio: Gardner, 1987.
2. Rosenberg, J., A History of Numerical Control 1949-1973, DoD DAHC-1572-C-0308, 1973.
3. Ogden, H., "Mechanical Design Aspects of Electronically Controlled Machine Tools," Proc. 4th Intl. MTDR Conf., 1963, p. 37. Oxford: Pergamon Press, 1963.
4. Leathan-Jones, B., Introduction to Computer Numerical Control. London: Pitman, 1986.
5. Koren, Y., Computer Control of Manufacturing Systems. New York: McGraw-Hill, 1983.
6. Pressman, R. and Williams, J., Numerical Control and Computer Aided Manufacturing. New York: Wiley, 1977.
7. Wick, C., "Advances in Machining Centers," Manufacturing Engineering, Oct. 1987, p. 24.
8. Koenigsberger, F., "Trends in the Design of Metal Cutting Machine Tools", CIRP Annals, Vol. 23, No. 2, 1974, p. 255.
9. Drayton, D. E. et al., "Automatically Programmed Tools," Numerical Control Programming Languages, Proc. 1st Intl. IFIP/IFAC PROLAMAT Conf., 1969. Amsterdam: North-Holland, 1970.
10. Shah, R., NC Guide-Numerical Control Handbook, 2nd end. Zurich, Switzerland: NCA Verlag, 1979.
11. ANSI Standard ANSI X3.37-1980: Programming Language APT. New York: American National Standards Institute, 1980.
12. Lange, K. et al., "Contribution of Numerical Control to the Development of Metal Forming Processes," CIRP Annals, vol. 31, No. 2, 1982, p. 511.
13. 1986-1987 Economic Handbook of the Machine Tool Industry. McLean, Va.: National Machine Tool Builders' Association, 1987.
14. Jaikumar, R., "Postindustrial Manufacturing," Harvard Business Review, Nov.-Dec., 1986, No. 6, pp. 69-76.
15. Mazak CAM-CNC system: Mazatrol CAMM-2 [manufacturer's catalogue]. Oguchi-cho, Japan: Yamazaki Machinery Works, 1983.
16. MC-DC Series Vertical Machining Center [manufacturer's catalog No. M0984-DC]. Fukui, Japan: Matsuura Machinery Corp.
17. Welch, A., "NC Verification Using Computer Simulation," CASA/SME paper MS80-233, Computer and Automated Systems Association/Society of Manufacturing Engineers, 1980.

18. Shpitalni, M. and Lemaistre, C., "The Problem of NC Program Verification: Analysis and Approaches," Computer Integrated Manufacturing, M. Martinez, et al., ed. New York: American Society of Mechanical Engineers, 1983.
19. Milner, D. A. and Vasiliou, V. C., Computer-Aided Engineering for Manufacture. New York: McGraw-Hill, 1987.
20. ANSI Standard (ANSI Y14.5M-1982): Dimensioning and Tolerancing. New York: American Society of Mechanical Engineers, 1983.
21. EIA Standard RS-267-B: Axis and Motion Nomenclature for Numerically Controlled Machines (ANSI/EIA RS-267-B-83). Washington, D.C.: Electronic Industries Association, June 1983.
22. ISO Standard 841-1974: Axis and Motion Nomenclature for Numerical Controlled Machines. Geneva, Switzerland: International Organization for Standardization, 1974.
23. EIA Standard RS-244-B: Character Code for Numerical Machine Control Perforated Tape. Washington, D.C.: Electronic Industries Association, Oct. 1976.
24. EIA Standard RS-358-B: Subset of American National Standard Code for Information Interchange for Numerical Machine Control Perforated Tape (ANSI/EIA RS-358-B-80). Washington, D.C.: Electronic Industries Association, Feb. 1979.
25. Childs, J. J., Principle of Numerical Control, 3rd ed. New York: Industrial Press, 1982.
26. FANUC System 6M-Model B Operator's Manual. Elk Grove Village, III.: General Numeric Corp.
27. FANUC Systems 6T Operator's Manual, Elk Grove Village, III.: General Numeric Corp.
28. Chang, C.-H., Lecture Notes for Course MANE 195A: "Numerical Control Manufacturing Machinery Laboratory," Manufacturing Engineering Program, University of California, Los Angeles, 1985.

Not: 21, 23 ve 24 no.lu referanslarda yer alan çalışmalar Electronic Industries Association, 2001 Eye St., N.W., Washington, D.C. 20006, adresinden 11 ve 20 no.lu referanslarda belirtilen çalışmalar ise American National Standards Institute, Inc., 1430 Broadway, New York, NY 10018'den satın alınabilir.

Kısım 2

APT Dilinde Bilgisayar Destekli NC Programlama

İnsanların CNC tezgahlarıyla bilgi iletiminde kullandıkları dil, kelime adres formatında NC kodlarından oluşmaktadır. Bölüm 2'deki açıklamalardan açıkça anlaşılmakta olduğu üzere, bir NC programı yazmak için kesici konumlarının hassas koordinatlarına gerek vardır. NC kodlarında programlama yapmanın iki dezavantajı mevcut olup; bunlardan birincisi, çoğu parça için kesici konumu koordinatları kolayca hesaplanamaz. Dahası elle yapılan hesaplama hem çok zaman alır, hem de hata şansı yüksektir. İkincisi, NC kodlarında elle yazılmış bir NC programı elle NC tezgahına ya da bir şerit-hazırlama cihazına girilmeli veya bilgisayara elle anahtarlanmalı ve oradan NC denetleyicisine gönderilmelidir. NC programları genellikle yüzlerce ya da binlerce deyimden oluştuğu için, elle giriş çok zaman alır ve saptanması güç yazım hatalarına yol açar. NC kodlarına, çok yinelemeli çevrim komutları, ayna komutları (freze kesicisinin bir ekseninde simetrik yansımaları), ve (değişken programlama niteliğine sahip NC denetleyicileri için) basit bir takım hesaplama ve karar deyimleri gibi çeşitli fonksiyonlar ekleme yolundaki çabalara rağmen bu durum pek değişmemiştir. Dolayısıyla freze kesicisi yolunu parça geometrisi (iş parçasının profili) esasına dayalı olarak tanımlamaya olanak sağlayan hesaplama yordamları ve karar deyimlerinin NC programına dahil edilmesini mümkün kılan yüksek düzeyli bir NC dili gerekmektedir. Programcıların NC programlama hatalarını bulmalarına yardımcı olması için de hata ayıklama özelliği gereklidir.

İyi tasarlandıkları ve yukarıda belirtilen niteliklere sahip olduğu düşüncesinden hareketle, varolan yüksek düzeyli bir dille, örneğin FORTRAN ya da PASCAL kullanılarak, NC programları yazılabileceği ileri sürülebilir. NC programlarını herhangi bir yüksek düzeyli dil kullanarak yazdığımızda ortaya çıkabilecek temel sorunlar şunlardır:

1. Bir kesici takım yolunu belirlemek için kullanılan geometrik cisimleri tanımlama güçlüğü
2. Kesici takım konumlarını (koordinatlarını) ya da yolunu hesaplama güçlüğü
3. Tezgah işlemini tamamlamak için gerekli işleme özelliklerini tanımlama güçlüğü

4. Kesici takımın hareket ve işleme özelliklerinin makine tarafından anlaşılabilir bir dilde (örneğin NC kodlarıyla) çıkışını alabilme güçlüğü.

Genel amaçlı yüksek-düzeyle dillerde NC programlaması mümkündür, ama aşağıdaki nedenler yüzünden hantal ve uygunsuzdur. Freze kesicisi yolu, bir dizi geometrik cisimden oluşan iş parçasının profili temel alınarak belirlenir. Freze bıçağı konumlarının (koordinatlarının) hesaplanması bir geometrik cismin diğerleriyle ya da diğerleriyle kesişimlerini bulmak ve kesici takım uç yarıçapı ve izin verilen son işleme payına ait telafi değerini hesaplamaktan ibarettir. Bu hesaplamalar bazen oldukça karmaşıklaşır ve somut bir matematik bilgisi gerektirir. Söz konusu hesaplama işlemlerinin geniş kapsamlı bir matematik eğitime sahip olmayan NC programcıları tarafından da kolaylıkla yapılabilecek şekilde basitleştirilmiş olması gerekir. Diğer taraftan, bir NC programının yazımı, geometrik cisimleri tanımlamak ve sonuçları, kullanılacak NC takım tezgahınca kabul edilebilir bir formatta (yani NC kodlarında) çevirmek ve çıktı almak için büyük iş gerektirir. Dolayısıyla geometrik cisimlerin tanımını basitleştirebilecek ve NC programcılarını karmaşık hesaplama yordamları tasarlamaktan çeviri ve çıktı yordamları yazmaktan kurtaracak özel bir yüksek-düzeyle dil gereklidir.

APT (Otomatik Programlanmış Takım), yukarıdaki sorunları çözmek üzere tasarlanmış yüksek düzeyli bir NC dilidir. Dili, kullanıcı için daha sıcak hale getirmek için İngilizce'ye benzer sözcükler kullanılmıştır. APT'de, parça işleme programlarını yürütebilecek bir programa (örneğin IBM APT-AC [otomatik programlanmış takım gelişmiş çevresel işleme] sayısını kontrol işlemcisine) ihtiyaç vardır. İşlemci, APT geometrik tanım deyimlerine göre gerekli geometrik ortamı oluşturur ve APT dilindeki kesici takım hareket deyimlerine göre kesici takım konumlarını ve yolunu hesaplar. İşleme koşullarıyla ilgili hesaplama sonuçları ve deyimler, APT programınca tanımlanmış sırada (örneğin CLDATA-kesici eksen verileri veya kesici yerleşim verileri-gibi) çıktı alınır ve sonrada bu veriler, kullanılacak sayısal kontrollü tezgaha özgün bir sonişlemci tarafından, tezgahın anlayabileceği NC kodlarına çevrilir.

APT, tezgah işlemleri için tasarlanmış bir dildir. Dolayısıyla, hatasız APT programları yazmak için metal işleme konusunda somut bilgiye sahip olmak şarttır. APT programlama dilinin kendisini tam olarak anlamının yanısıra kullanılacak son işlemcilere dair bilgi de gerekmektedir.

APT standartlaştırılmış olmasına karşın, çeşitli bilgisayar sistemlerinde hala belli farklılıklar bulunmaktadır. Aşağıdaki inceleme, APT programının, IBM APT-AC sayısal kontrol işlemcisinin 1. versiyon ve 3. modifikasyon düzeyi tarafından işleneceği varsayımıyla IBM bilgisayar sistemi için geçerlidir.*

*Aşağıdaki incelemede, IBM APT-AC sayısal kontrol işlemcisi; APT-AC NC işlemcisi, ya da basitçe NC işlemcisi olarak anılacaktır.

İyi tasarlanmış ve yukarıda saydığımız niteliklere sahip olmaları nedeniyle, NC programları yazımında FORTRAN ya da PASCAL gibi mevcut bir yüksek düzeyli dille NC programları yazılırken şu sorunlarla karşılaşılabilir: Rehber (NC programcısı) için kesici takım hareket yolunu önceki şekilde belirlemek (tanımlamak) daha kolaydır.

Bölüm 3

APT Dilinin Elemanları

Herhangi bir bilgisayar programı gibi, APT parça işleme programı da, deyim adı verilen ve bir bilgisayarın istenilen işlem sırasını izlemesini sağlayan, düzenli olarak listelenmiş bir komutlar kümesidir. Aşağıda, APT deyimlerine ait birkaç örnek verilmiştir:

```
MACHIN/GN5CC,9,OPTION,2,0
PT0=POINT/0,0,1,5
A3)SPINDL/500,CLW
```

Bu örneklerden de görüleceği gibi, her deyim aşağıdaki elemanların bir veya bir kaçından oluşmaktadır:

- Kelimeler (örneğin; MACHIN, OPTION, POINT)
- Simgeler (örneğin; GN5CC ve PT0)
- Sayılar (örneğin 9, 2, 1.5)
- Noktalama ve sınırlayıcılar (örneğin kesme, virgül)
- Deyim etiketi (örneğin; parantez kapanışından önceki A3 karakterleri)

Simgeler, kelimeler ve deyim etiketleri İngilizce harflerinden (A, B, ..., Z büyük harfle) ya da İngilizce harfler ve sayısal basamaklarla (0, 1, 2, ..., 9) yapılır. Sayılar, sayısal ondalık basamaklar olarak tanımlanır. APT'de noktalama simgesi ve sınırlayıcı olarak kullanılan karakterler Tablo 3-1'de gösterilmektedir. Aşağıdaki incelemede bir alfasayısal karakter ya basamak olarak ya da İngiliz harfi olarak tanımlanmaktadır; bir alfasayısal dizi, bir ya da daha çok alfasayısal karakterden oluşur.

TABLO 3-1 APT DİLİNDE KULLANILAN KARAKTERLER

Karakter	Anlam ve Kullanım
	Boş ya da sıfır karakter (anlamlı alfasayısal metin dışında NC işlemcisi tarafından yok sayılır)
\$	Dolar işareti - deyim bir sonraki satırda devam ediyor
\$\$	Çift dolar işareti - deyim sonu
;	Noktalı virgül, aynı satırdaki iki deyim böler
: veya)	İki nokta veya parantez, önceki alfasayısal karakterlerin bir deyim etiketi oluşturduğunu gösterir
()	Parantez aç ve kapa - içiçe veya parantezli ifade
' '	Bir çift apostrof - alfasayısal anlamı sınırlayıcı
*	Yıldız - çarpma
**	Çift yıldız - üst alma
+	Artı - pozitif işareti ya da toplama
-	Eksi - negatif işareti ya da çıkarma
.	Nokta - ondalık nokta
/	Kesme - bölme ya da ana-alt sözcük ayracı
=	Eşittir - operatörü atar
,	Virgül - sözcük veya sayı ayracı
" "	Tırnak işareti kompleks bir MACRO değişkeni özelliğini sınırlar
	İki çizgi veya vuruş - kanonik parametrenin hariç sayılması
A,B,...,Z	İngiliz alfabesi (büyük harf) sözcükleri, simgeleri, değişkenleri ve deyim etiketlerini kurmak için kullanılır.
E veya D	İngiliz alfabesi (büyük harf) - bir sayıyı üstsel formatta tanımlamak için kullanılır
PI	simgesi (3.14159265359879310)
0,1,...,9	Sayısal karakterler - bir sayı, simge ya da deyim etiketi oluşturmakta kullanılır

Kullanılan karakterler Tablo 3-1 de gösterilmektedir. Aşağıdaki incelemede bir alfa- sayısal karakter ya basamak olarak ya da İngiliz harfi olarak tanımlanmaktadır; bir alfasayısal dizi, bir ya da daha çok alfasayısal karakterden oluşur.

3.1 APT KELİMELERİ

APT kelimeleri birden altıya kadar alfasayısal karakter dizisinden oluşur. Beş kategori altında incelenebilir (Tablo 3-2):

- Geometrik cisimleri tanımlamak için kullanılanlar
- Matematiksel işlemleri tanımlayanlar
- Kesici takım hareketini tanımlamak için kullanılanlar
- (NC işlemcisine CLDATA'yı basması için komut veren) CLPRNT ya da (son işlemcinin yürümesini bastıran) NOPOST gibi bilgisayar işlem modlarını tanımlayanlar
- Çoğu; tezgahta işleme özellikleri, işaretler ya da açıklamalarla ve benzeri unsurlarla ilgili olan ve doğrudan sonişlemci tarafından işlenen son işlemci sözcükleri

TABLO 3-2 SEÇİLMİŞ APT SÖZCÜKLERİ

Geometrik Cisim Tanımı	Matematiksel İşlem	Kesici Takım Hareketi	Bilgisayar İşlem Modu	Sonışılmalı Kelimesi
ABSLTE	ABSF	AUTOPS	CALL	ARCSLP
ARC	ACOSF	CUT	CANON	AUXFUN
AT	ANGLF	DNTCUT	CLPRNT	BRKCHP
ATANGL	ASINF	FROM	CONTIN	CCLW
ATTACH	ATANF	GO	COPY	CLEARP
AVOID	ATANGF	GOBACK	DO	CLRSRF
BISECT	CBRTF	GODLTA	FINI	CLW
CENTER	COSF	GODOWN	IF	COOLNT
CCLW	COTANF	GOFWD	INSERT	CUTCOM
CIRCLE	DISTF	GOLFT	JMPTO	CUTTER
CLW	DOTF	GORGT	LOOPST	CYCLE
CONE	EXPF	GOTO	LOOPND	DEEP
CONST	INTGF	GOUP	MACHIN	DELAY
CROSS	LNTHF	INDIRP	MACRO	DRILL
CYLNDR	LOGF	INDIRV	NOPOST	DWL
DECR	LOG10F	NOPS	PARTNO	END
DELTA	MAXIF	OFFSET	PPRINT	FEDRAT
ELLIPS	MINIF	ON	PRINT	FLOOD
FUNOFY	MODF	PAST	REMARK	GOCLER
GCONIC	NUMF	POCKET	RESERV	GOHOME
HYPERR	SIGNF	PSIS	RESET	HIGH
IN	SINF	REFSYS	TERMAC	INCHES
INCR	SQRTF	TANTO	TITLE	INTOL
INTERC	TANF	THICK		IPM
INTOF	TYPEF	TLLFT		IPR
INVX	'EQ'	TLON		LEADER
INVY	'GE'	TLRGT		LOADTL
LARGE	'GT'	TRACUT		LOW
LCONIC	'LE'			MCHTOL
LEFT	'LT'			MIST
LINE	'NE'			MM
LINEAR				OFF
MATRIX				ON
MINUS				OPSKIP
NEGX				OPSTOP
NEGY				OPTION
NEGZ				ORIGIN
NORMAL				OUTTOL
NOX				PLABEL
NOY				PREFUN
NOZ				RAPID
NUMPTS				RETRCT
OBTAIN				ROTABL
OMIT				ROTREF
OUT				SELCTL
PARLEL				SEQNO
PATERN				SETOOL
PERPTO				SPINDL
PLANE				STOP

Not: APT sözcüklerinin tam listesi için Ek A'ya bakınız.

TABLO 3-2 DEVAM

Geometrik Cisim Tanımı	Matematiksel İşlem	Kesici Takım Hareketi	Bilgisayar İşlem Modu	Sonışılmalı Kelimesi
POINT				TAPKUL
POLCON				TOLER
POSX				UNITS
POSY				
POSZ				
PLUS				
PTNORM				
PTSLOP				
QUADRIC				
RADIUS				
RETAIN				
RIGHT				
RLDSUF				
RTHETA				
SLOPE				
SMALL				
SPHERE				
SPLINE				
STEP				
TABCYL				
TANTO				
THETAR				
THRU				
TIMES				
TORUS				
TRANPT				
TRANSL				
TRFORM				
TWOPT				
UNIT				
VECTOR				
XAXIS				
XCOORD				
XLARGE				
XSMALL				
XYPLAN				
XYZ				
YAXIS				
YCOORD				
YLARGE				
YSMALL				
YZPLAN				
ZAXIS				
ZCOORD				
ZLARGE				
ZSMALL				
ZSURF				
ZXPLAN				
3PT2SL				
4PT1SL				

Bir giriş satırının 1'den 6'ya kadar olan sütunlarında belirtilmesi gerekli olan ve sabit alanlı kelimeler adı verilen birkaç sözcük mevcut olup; bunlar PARTNO, REMARK, PPRINT, INSERT ve TITLE'dir.

IBM APT-AC NC işlemcisince kabul edilebilir çok sayıda (yaklaşık 600) kelime vardır. Bu kitapta, biz bunların en çok kullanılan (Tablo 3-2) ikiyüz kadarını göstereceğiz. Ek A, IBM APT-AC NC işlemcisinin kabul ettiği kelimelerin listesini referans amacıyla vermektedir.

Matematiksel işlemleri tanımlayanların dışında kalan kelimeler iki gruba ayrılabilir; *ana kelimeler* ve *ikincil kelimeler*. Ana kelimeler, kesme işaretinden önce yerleştirilir ve geometrik cisim tipini, gerekli bilgisayar işlemi, tezgah işleme özelliklerini, veya kesici takım hareketinin tipini, yönünü ve kesici takım profil ilişkisini belirler. İkincil kelimeler genellikle düzelticilerdir ve geometrik cisimleri, kesici takım hareketini ya da tezgah işleme özelliklerini ana kelimelerin belirlediği aralıkta tanımlamak için kullanılır. Kelimelerin anlamı ve kullanışı sonraki bölümlerde ayrıntılı olarak ele alınacaktır.

3.2 SEMBOLLER

Bir sembol, en azından birini sayısal olmayan birden altıya kadar alfasayısal karakterden oluşur. Programın ilerleyen kısmında kolayca başvurulabilecek bir geometrik cisme, matematiksel ifadeye, altprograma ya da sayıya atanır. Örneğin

L1=LINE/0,0,0,1,1,1

deyiminde, (0,0,0) ve (1,1,1) koordinatlarındaki iki noktadan geçen doğru tanımlanmakta ve bu doğruya L1 sembolü atanmaktadır.

APT'de tanımlanan herhangi bir kelimenin simge olarak kullanılmasına izin verilmez. APT'de yüzlerce kelime olduğu ve bunların pek azı sayısal karakter içerdiği için, her sembolde en az bir sayısal karakterin bulunması tavsiye edilir. Boşluklar ve noktalama işaretleri bir sembol oluşturamaz.

3.3 SAYILAR VE SKALERLER

Bir APT programındaki bütün sayılar APT-AC NC işlemcisi tarafından gerçek sayılar olarak işlenir. Dolayısıyla 5 sayısı aşağıdaki şekillerin herhangi biriyle ifade edilebilir:

5

+5

5.0
0.5E1 (örneğin, 0.510 1)
.5E+1 (örneğin, 0.510 1)
50.E-1 (örneğin, 5010 1)
.05E2
0.5D1 (örneğin, 0.510 1)
0.05D+2

ve bunun gibi.

Burada D ve E, sayıyı üstsel biçimde belirlemek için kullanılmıştır aynı anlama ve kullanışa sahiptir. E ve D singelerinden önceki sayının ondalık basamağı olması gerektiğine dikkat ediniz; aksi takdirde, sayıyı tanımlayan bütün karakter kümesi değişken bir sembol olarak görülür.

Bir sayıda izin verilen en büyük basamak sayısı 75'tir. Bir APT programında belirtilebilen sayılar 10^{-75} 'ten 10^{75} 'e kadar değişebilir; ancak, işlem sırasında NC işlemcisi sayının ondalık noktasından sonra sadece onaltı ondalık basamağı anlamalı kabul eder.

Bir tamsayı bağımsız bir sayı sınıfı olarak görülmez. Eğer bir sayı indis olarak kullanılıyorsa (bu durumda bir tam sayıya ihtiyaç vardır), sadece bu sayının tamsayı kısmı kullanılır. Pozitif bir sayı artı işaretiyle ve artı işaretsiz olarak belirtilebilir; negatif sayıdan önce eksi işareti olmalıdır.

Bir açının büyüklüğünü belirten bir sayı, derece cinsinden, derecenin ondalık kesirleri cinsinden ya da her ikisi olmalıdır. Bir açı saat yönünün aksi yönde ölçülüyorsa pozitif, saat yönünde ölçülüyorsa negatiftir (Şekil 2-3'e bakınız).

Aşağıda "skaler" terimi, tek değerli bir varlığı tanımlamak için kullanılmaktadır. Bu terim; azaltıldığı zaman tek değerli olan bir ifade, sembol ya da sayı olabilir.

3.4 NOKTALAMA İŞARETLERİ VE SINIRLAYICILAR

3.4.1 Boşluk Karakteri

Boşluk karakteri APT-AC NC işlemcisi tarafından yok sayılır ve aşağıdaki durumlar dışında bir cümle içinde her yerde belirtilebilir:

- Anlamlı bir metinde bir çift apostrof tarafından sınırlanmış bir boşluk, anlamlı metnin bir parçası olarak ele alınır.
- Bir boşluk, sabit alanlı bir sözcükten önce veya onun içine yerleştirilebilir. (örnek PARTNO, INSERT, REMARK, PPRINT veya TITLE).

- \$ ya da \$\$ işaretinden veya sabit alanlı bir sözcükten sonra yerleştirilen bir boşluk açıklayıcı metnin parçası olarak görülür.

Normalde, bir boşluk ya da boşluklar, NC programının okunurluğunu arttırmak için tanımlanır. Karışıklığa neden olacağı için deyim etiketinde boşluk kullanılmaması önerilir.

Örnek 1

Boşluklar, şu APT programının okunurluğunu arttırmak için kullanılmaktadır:

```
.....
.....
L1 = LINE/PT1, PT2
C(1) = CIRCLE/CENTER, PT3, RADIUS, 1.0
PL(1) = PLANE/0,0,1,3
.....
```

```
TLLFT, GOLFT/L1, TO, C(1)
GOLFT/C(1), TO, L2
.....
.....
```

Örnek

Aşağıdaki deyimlerde, boşluk ya da boşluklar bir açıklama ya da anlamlı metnin bir parçası olarak dikkate alınacaktır:

```
PARTNO A 3-D PROFILE CUTTING PROGRAM
REMARK TOOL DIAMETER = 0.75 IN.
REMARK/TOOL DIAMETER = 0.75 IN.
```

3.4.2 \$ (Dolar İşareti)

Bu işaret, kendisinden önce gelen deyim bir sonraki satırda devam ettiğini gösterir. Aynı satır üzerinde bu sembolü izleyen karakterler NC işlemcisi tarafından yok sayılır ve açıklama olarak kullanılır. Ancak, bu sembol sabit alanlı bir sözcükten sonra, açıklama metninin bir parçası olarak ele alınır. Dolar işareti, bir deyim 72 karakterden (bir satırda girilebilecek maksimum karakter sayısı) oluştuğu zaman kullanılır.

Örnek 1

```
L1 = LINE/P1,LEFT,$
TANTO.C1
```

deyimi

```
L1 = LINE/P1,LEFT,TANTO,C1
```

ile aynıdır.

Örnek 2

Aşağıdaki deyimde dolar işaretinden sonra gelen karakterler bir açıklama oluşturmaktadır:

```
C1 = CIRCLE/XLARGE,L1,$ THIS CIRCLE IS TANGENT TO THREE LINES L1, L2, L3
      YLARGE,L2,XSMALL,L3
```

3.4.3 \$\$ (Çift Dolar İşareti)

Çift dolar işareti (sabit alanlı bir sözcükten sonra yerleştirilmediği sürece) bir deyim sonunu temsil eder. Aynı satır üzerinde bu işareten sonraki bilgi açıklama olarak algılanır. Çoğu yüksek düzeyli dilde olduğu gibi, APT dilindeki bir deyim, bir deyim devamı işareti (\$) belirtilmedikçe bir satırı kopyalayacağı düşünülür. Dolayısıyla deyim sonu işaretinin, \$\$, kullanılması isteğe bağlıdır; sadece aynı satırda bir açıklama belirtilecekse kullanılır.

Örnek

Aşağıda deyim, bir P1 noktasını, C1 çemberinin merkezi olarak açıklayan bir ifade ile tanımlamaktadır:

```
P1=POINT/1.0,2.0,3.0$$$ THIS IS THE CENTER OF CIRCLE C1
```

3.4.4 ; (Noktalı Virgül)

Bu işaret, aynı satır üzerinde iki deyimi bölmek için kullanılır. Ancak, aynı satırda tek bir dolar işaretinden veya sabit alanlı bir kelimeden sonra kullanılması aynı etkiye sahip değildir; bu durumda açıklama metninin parçası olarak düşünülür. Eğer bir çift apostrof tarafından sınırlanmış bir alfasayısal dizi içinde noktalı virgül belirtiliyorsa, sanıldığı gibi alfasayısal dizinin parçası olmaz, ama deyim iki parçaya bölerek deyim sonu işareti işlevi görür.

Örnek 1

```
INDIRV/V1;GO/S1,S2
```

giriş satırı, iki ardışık satırda belirtilen şu iki deyime eşdeğerdir:

```
INDIRV/V1
GO/S1,S2
```

Örnek 2

INDIRV/V1\$\$V1 IS SHOWN ON THE PART DRAWING;GO/S1,S2

giriş satırı, noktalı virgülle, içinden bir tanesi aşağıdaki açıklama metnine sahip olmak üzere şu iki deyim bölünür:

INDIRV/V1\$\$V1 IS SHOWN ON THE PART DRAWING GO/S1,S2

Örnek 3

Aşağıdaki deyimlerin C1, C2 ve C3 çemberlerini tanımlamak için kullanılması amaçlanmaktadır:

C1 = CIRCLE/YSMALL,L1,YLARGE,L2,RADIUS,R\$FIRST CIRCLE;C2 = CIRCLE/1.,2.,1.5 C3 = CIRCLE/1.,2.,1.5

Ancak, ilk satırda tek dolar işaretinden sonraki giriş ihmal edilmiştir. NC işlemcisi iki satırı bir deyim gibi okur; sözdizim olarak yanlış olan bu deyim şöyledir:

C1 = CIRCLE/YSMALL,L1,YLARGE,L2,RADIUS,RC3 = CIRCLE/1.,2.,1.5

Örnek 4

Aşağıdaki deyimde sabit alanlı REMARK sözcüğünden sonraki noktalı virgül açıklama metninin parçasıdır:

REMARK TOOL DIAMETER 1/2 IN FOR OUTER PROFILE; 1/4 IN FOR INNER PROFILE

3.4.5 : (İki Nokta Üstüste) veya (Kapama Parantezi)

İki nokta üstüste bir önceki alfasayısal dizinin bir deyim etiketi olduğunu göstermek için kullanılır. Açma paranteziyle kullanılmadığı sürece kapama parantezi de üstüste iki noktayla aynı anlama gelir. Bir deyim etiketi deyim başlangıcında belirtilmelidir.

Örnek

A1 ve 1B aşağıdaki iki deyim deyim etiketleridir:

A1)CALL/C1,P = 1.0,Q = 2.0

1B:JUMPTO/3C

3.4.6 () (Açma ve Kapama Parantezi)

- Açma ve kapama parantezleri şu amaçlar için kullanılabilir:

Örneğin

K = TANF(B)

gibi, B'nin tanjant fonksiyonu argümanı olduğu matematiksel fonksiyon argümanlarını kapamak için;

Örneğin

M(1) = MATRIX/XYROT,15

gibi, karakter 1'in M sembolünün alt indisi olduğu, bir sembol ya da değişkenin alt indis veya alt imini kapamak için;

- IF koşullu dallanma deyiminin argümanını kapamak için. Argüman bir değişken, matematiksel bir ifade ya da ilişkisel bir ifade olabilir.

Örnek 1

Kapanan A argümanı bir değişkendir:

IF(A)1A,2A,3A

Örnek 2

Kapanan A-B argümanı matematiksel bir ifadedir:

IF(A-B)C1,C2,C3

Örnek 3

Kapanan A'EQ'B argümanı ilişkisel bir ifadedir:

IF(A'EQ'B).JUMPTO/A1

- JUMPTO koşullu dallanma deyiminin deyim etiketleri listesini kapamak için:

JUMPTO/(C1,C2,C3,C4),N

burada C1, ..., C4 deyim etiketleridir.

- İççe tanımlı kapamak için.

Örnek

L1 doğrusunun, P1 ve P2 noktalarıyla aşağıdaki gibi tanımlanması amaçlanmaktadır:

$$L1 = \text{LINE}/P1,P2$$

Ancak, P1 programda henüz tanımlanmamış olabilir. Bu sorunu çözmek için aşağıdaki iççe tanım kullanılabilir:

$$L1 = \text{LINE}/(P1 = \text{POINT}/1,0,0,2,0),P2$$

- Bütün olarak ele alınan bir şeyi kapamak veya diğerlerine göre işlem önceliği olan bir matematiksel ifadenin bir kısmını kapamak için.

Örnekler

$$A = B*(-C)**(-2.5)$$

$$A = (B*(C/D)+F)**G$$

3.4.7 / (Kesme İşareti)

Kesme işareti, bir deyimdeki ana elemanları ayırmak için veya aritmetik bölme sembolü olarak kullanılır.

Örnek 1

Aşağıdaki deyimde kesme işareti, geometrik cismin türünü ve parametreleri tanımlayan kelimeleri ayırmak için kullanılmaktadır.

$$\text{POINT}/1,0,2$$

Örnek 2

Aşağıdaki deyimde kesme, aritmetik işlem (bölme) amacıyla kullanılmıştır.

$$K = (A+B)/(C+D)$$

3.4.8 = (Eşittir İşareti)

Eşittir işareti, bir geometrik cisime (Kısım 3.2'deki örneğe bakınız) ya da altprograma (APT'de MACRO adı verilir) isim atamak için kullanılır; ya da bir değişkene bir değer atamakta kullanılabilir.

Örnek

Aşağıdaki deyimler sırasıyla, tanımlanmış altprograma (MACRO) K1 ismini ve X1 değişkenine 2.75 değerini atamaktadır:

$$K1 = \text{MACRO}/A,B,C$$

$$X1 = 2.75$$

3.4.9 + (Artı İşareti), - (Eksi İşareti), * (Yıldız) ve ** (Çift Yıldız)

Artı ve eksi işaretleri ve tek yıldızla çift yıldız sırasıyla toplama, çıkarma, çarpma ve üst alma matematiksel işlemlerini temsil etmektedir. Kesme işaretiyle birlikte bütün basit aritmetik işlemci kümesini oluştururlar. "+" ve "-" işaretleri aynı zamanda sırasıyla bir pozitif ve bir negatif sayıyı belirtmek için de kullanılır. İşaretsiz bir sayı pozitif olarak değerlendirilir.

Örnekler

$$A = B + C - M/N$$

$$F = (G**(-H/2))**I$$

NC işlemcisi aşağıdaki öncelik sırasını takip ederek, parantezsiz bir ifadeyi, yani içinde parantez kullanılmayan bir ifadeyi değerlendirir:

Birinci öncelik:	Üst alma (**)
İkinci öncelik:	Çarpma (*) ve bölme (/)
Üçüncü öncelik:	Toplama (+) ve çıkarma (-)

Bilgisayar parantezsiz bir ifadeyi soldan sağa üç kez tarar. Önce üst alma işlemlerine bakar ve sonucu (veya sonuçları) hesaplar; sonra çarpma ve bölme bakar; son olarak toplama çıkarmaları yapar. NC işlemcisi parantezlerin geleneksel şekilde kullanımına da olanak sağlamaktadır; yani parantezler bütün aritmetiksel işlemlere göre öncelik taşır ve iç taraftaki parantez önce değerlendirilir.

Örnek 1

$$A = B/C + D * F ** G$$

deyimi aşağıdaki klasik matematiksel ifadeye özdeşleştir:

$$A = \frac{B}{C} + D \cdot F^G$$

Örnek 2

$$A = \frac{B}{C} + (D \cdot F)^G$$

matematiksel ifadesi programda belirtilecek ise, deyim şu şekilde yazılmalıdır.

$$A = B/C+(D*F)**G$$

Örnek 3

Hatalar ve karışıklıklar aynı öncelik sırasına sahip iki ardışık işletmeni bir çift parantezle ayırarak önlenebilir. Aşağıdaki tabloda bazı APT deyimi örnekleri ve bu deyimlerin anlamları verilmektedir:

APT DEYİMİ	ANLAMI
$A = B * C / D$	$A = \frac{B \cdot C}{D}$
$A = B * (C / D)$	$A = B \cdot \frac{C}{D}$
$A = B / C / D$	$A = \frac{B}{C / D}$
$A = B / C / D$	$A = \frac{B}{C / D}$

3.4.10 . (Nokta ya da Ondalık Noktası)

Nokta bir tamsayıyı sayının kesirli kısmından ayırmak için kullanılır (örneğin 3.52, 0.002).

3.4.11 , (Virgül)

Kesme işareti ve paranteze ek olarak, virgüller de bir deyimden öğelerini ayırmak için kullanılır.

Örnekler

MACHIN/LATH1,6,OPTION,2,0
TLRGT,GORGT/(L1=LINE/P1,P2),ON,L2

3.4.12 || (Çift Çizgi)

Çift çizgi işareti geometrik bir cisim için kanonik biçimde bir öge çıkarmak için kullanılır. Örneğin C1 çemberi için kanonik biçim $x, y, z, 0, 0, 1, r$ 'dir (Ek B'ye

bakınız); burada x, y ve z merkezin koordinatları, $0, 0, 1$ normalin, çemberi kapsayan düzleme dik olan doğrunun yönsel katsayıları; ve r 'de yarıçaptır. C1 çemberinin merkezini P1 noktası olarak tanımlayan deyim aşağıdaki gibidir:

$$P1 = \text{POINT}/C1|1|,C1|2|,C1|3|$$

Bu deyimde çift çizgi içine alınan 1, 2 ve 3 sayıları kanonik formun sırayla birinci, ikinci ve üçüncü öğelerinin kullanılmasını gerektirdiğini göstermektedir.

3.4.13 " (Bir Çift Tırnak)

Bir çift tırnak anlamı bir karakter dizisini kapamak için kullanılır (Kısım 3.4.1'de örnek 2'ye bakınız).

3.4.14 " " (Bir Çift İkili Tırnak İşareti)

Bir çift ikili tırnak işareti bir MACRO (altprogram) değişkeni için karmaşık bir özelliği kapamak için kullanılır.

Örnek

A1 = MACRO/C1,C2,C3=...TLLFT,GOLFT...

3.5 DEYİM ETİKETLERİ

Eğer bir deyimde, diğer deyimlerde başvurulacaksa, bir etiketi olmalıdır. Bir deyim etiketi, her biri sayısal ya da alfabetik olabilen birden altıya kadar alfasayısal karakterden oluşur. Etiket, deyimden başlangıcına yerleştirilmeli ve kapama parantezi ya da iki noktayla kapatılarak deyimden ayrılmalıdır.

Örnekler

1A)CUTTER/0.5,0
15:FEDRAT/2.0
A31)GOTO/P1
AN1CY2)GODLTA/-0.2

Eğer başka bir deyimde atıfta bulunulmayacaksa, bir deyimde etiket atamaya gerek yoktur.

3.6 DEYİM BOYUTU SINIRLAMA

Bir deyim yukarıda açıklanan bir veya birkaç öğeden oluşmakta olup; bir APT deyiminde belirtilebilecek en yüksek öğe sayısı 4092'dir. Örneğin,

A1)GOTO/(POINT/0,0,0)

deyimi 11 öğeden; yani iki sözcük (GOTO ve POINT); altı noktalama işareti (iki kesme, iki virgül ve açma kapama parantezleri); ve üç sayısal karakterden oluşmaktadır. Deyim etiketinin ve bu etikete ait kapama parantezinin (ya da üstüste iki noktanın) deyimden bileşeni olarak değil fakat bağlı bir parçası olarak dikkate alındığına dikkat ediniz. Dolayısıyla bunlar bir deyimden öğeleri sayılmaz.

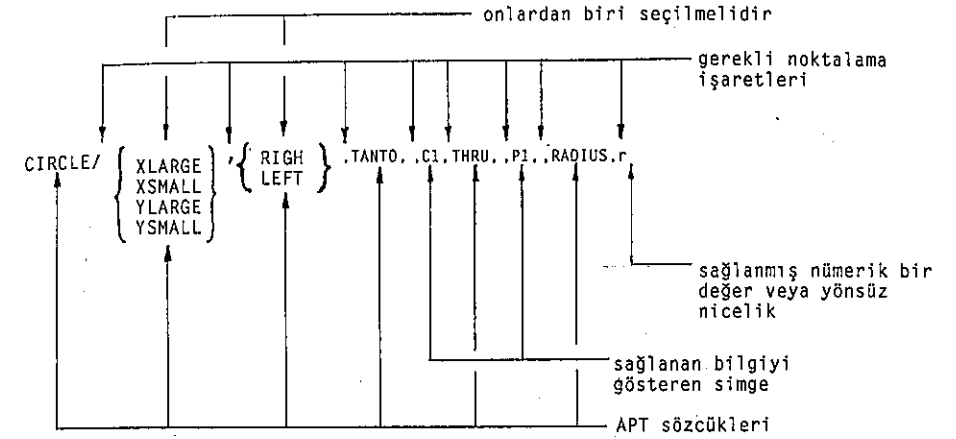
3.7 APT DEYİM FORMATI İÇİN GÖSTERİM (NOTASYON)

Bu kitapta APT deyim formatını açıklamak için aşağıdaki sembol ve kelime kuralları kullanılmaktadır:

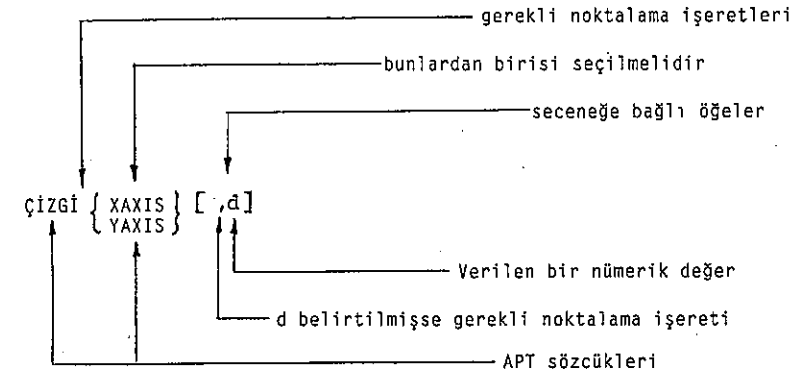
1. Bütün APT kelimeleri büyük harflerle basılır.
2. Sayısal bir değeri ya da programlamada olmaması gereken bir skalesi niceliği temsil eden bir kelime, küçük İngilizce harfler, bir veya daha çok basamaklı ya da basamaksız karakterler ile (örneğin, r, r1, r12) gösterilir.
3. Sayısal değer veya skalar dışında kalan, programcı tarafından sağlanan bilgi, hem büyük yazılmış İngilizce karakter(-ler) ve hem de basamak(lar)dan oluşmaktadır.
4. APT deyiminin bir öğesi için mümkün olabilecek seçenekler ayrıçlarla ayrılmıştır ve programlama sırasında bunlardan biri seçilmelidir. Ancak, ayrıçların kendileri deyimde belirtilmemelidir.
5. İsteğe bağlı öğe(-ler) (örneğin gerektiğinde ya da ihmal edildiğinde belirtilen öğe(-ler) parantez içine alınır(lar). Ancak parantezlerin kendileri deyimde belirtilmemelidir.
6. Bir deyimde tüm noktalama işaretleri ve sınırlayıcılar gereklidir ve bunların deyimdeki konumları değiştirilmemelidir.

Şekil 3-1'deki örnekler yukarıda açıklanan kuralları göstermektedir.

ÖRNEK 1:



ÖRNEK 2:



Şekil 3-1 Sembol ve kelimelerin kurallarını gösteren iki örnek.

Bölüm 4

Geometrik Öğelerin Tanımlanması

Kinematik açıdan bakıldığında, bir tezgah işleminin programlanması, esas itibarıyla tezgah kesicisinin işlenecek parça profili boyunca belirli bir sapma toleransı dahilinde hareket edecek şekilde yönlendirilmesinden ibarettir. Bu nedenle, iş parçasının profili, kesici takımının profil verme hareketi için yönlendirici bir faktördür. Genellikle bir parçanın profili; iki boyutlu işleme için doğru, çember yayı ise düzlem ve profilili yüzeylerden yapılır. Bu elemanların, bir kesici takımının (örneğin freze kesicisinin) hareketi belirlenmeden veya programlanmadan önce tanımlanmış olmaları kesinlikle şarttır.

APT bilgisayar dilinde, bir geometrik öğe, öğenin tipini gösteren birincil derecede önemli bir kelime ile bu kelimedenden sonra yeralan ve söz konusu varlığın konumunu, ölçüsünü, ya da sınırlama şartını belirten ikinci derecede önemli bir veya daha fazla kelime ya da parametre vasıtasıyla tanımlanır. Örneğin, merkezi (1.0, 2.0, 2.5) noktasında bulunan ve yarıçapı 3.0'e eşit olan bir daireyi tanımlayan ifade:

C1=CIRCLE/CENTER, 1.0, 2.0, 2.5, RADIUS, 3.0

şeklinde olabilir. Burada, C1 tanımlanan daireye verilen bir semboldür. CIRCLE birincil, CENTER ve RADIUS ikincil kelimelerdir. APT-AC NC işlemcisi; nokta, doğru, vektör, düzlem, daire, silindir, koni, elips, hiperbol, küre, taşınmış yüzey ve yuvarlak çıkıntı dahil çok sayıda geometrik öğenin tanımlanabilmesini mümkün kılar. Bu öğelerin çoğu için, farklı parametrelere sahip çok sayıda tanımlama formatı mevcut olup; bunların yardımıyla bilinen çeşitli şartlara dayalı bir geometrik öğenin tanımlanabilmesi sağlanmış olur.

Örneğin bir nokta, koordinatlarıyla verilen bir noktaya göre ilan konumu ile ya da verilen iki veya daha fazla geometrik varlığın kesişmesiyle tanımlanabilir. NC'li tezgahın işlemcisi bir çözümü (yani istenen noktayı) hesaplamak için seçilen şartı kullanır ve sonra da bilgiyi "Kanonik Biçim" olarak bilinen bir standart formata depo eder (Kısım 4.11 ve Ek B'ye bakınız). Verilerin bir şartlar kümesi için birden fazla çözüm mevcut olabilir. Böyle bir durumda, mümkün olabilecek tüm çözümler arasında en doğru olanını seçmek için ilave değiştiricilerin (modifikatörlerin) belirtilmesi gerekir. Bu değiştiriciler şunları içerir:

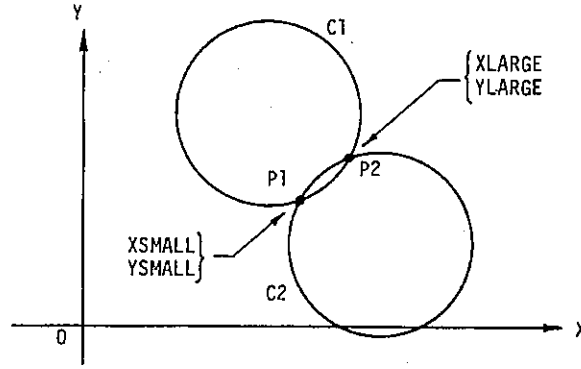
- Doğrultu değiştirici: XLARGE, XSMALL, YLARGE, YSMALL, ZLARGE, ZSMALL, POSX, POSY, POSZ, NEGX, NEGY, NEGZ, LEFT ve RIGHT; CLW, CCLW;
- Boyut değiştirici: LARGE ve SMALL
- İlişki değiştirici: IN (İç) ve OUT (Dış)

Örneğin X-Y düzlemindeki bir nokta, C1 ve C2 olarak verilen iki çemberin kesiştiği nokta olarak tanımlanabilir ve şu deyim (Şekil 4-1'e ve Tablo 4-3'deki No.4 deyim formatına bakınız) ile ifade edilebilir:

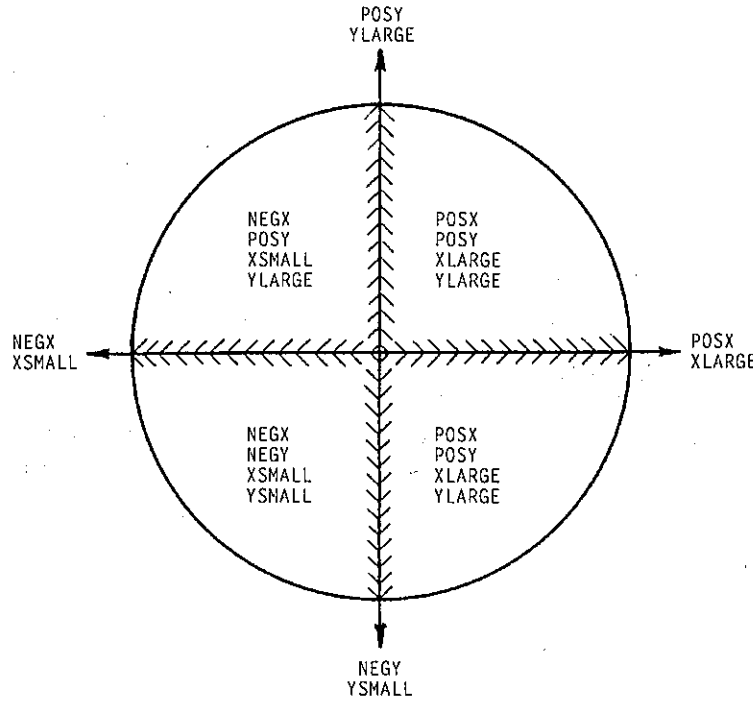
POINT / $\left. \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}$ INTOF, C1, C2

Genel olarak, verilen şartı sağlayan iki çözüm (nokta) vardır. Buna göre, istenen çözümü göstermek üzere ilgili doğrultu düzeltici seçilir. APT'de kullanılan doğrultu değiştiricilerin anlamları aşağıdaki gibidir (Şekil 4-2):

DOĞRULTU DEĞİŞTİRİCİ	ANLAMI
POSX, XLARGE POSY, YLARGE POSZ, ZLARGE	İstenen çözüm daha büyük x,y ve z koordinatlı olanıdır ya da mümkün olan diğer çözümle karşılaştırıldığında, ortam X,Y veya Z yönünde bulunur.
NEGY, XSMALL NEGY, YSMALL NEGZ, ZSMALL	İstenen çözüm daha küçük x,y veya z koordinatlı olanıdır ya da mümkün olan diğer çözümle karşılaştırıldığında, azalan X, Y veya Z yönünde bulunur.



Şekil 4-1 Mümkün olan iki çözüm arasında isteneni göstermek için doğrultu değiştiricilerin kullanımı. Bir doğrultu değiştirici istenen çözümün koordinatlarını diğer çözümlükilerle karşılaştırmak suretiyle seçilebilir. Dolayısıyla, P1 noktası için XSMALL veya YSMALL değiştiricisi seçilebilir; P1 noktası için XLARGE veya YLARGE geçerlidir.



Şekil 4-2 X-Y düzleminde doğrultu değiştirici seçimi. Dairenin merkezini başlangıç noktası veya referans pozisyonu olarak kullanmak suretiyle üç pozisyonuna veya tanımlanmış geometrik şekle doğru bir vektör çiziniz. Mümkün olan değiştirici seçenekleri, sonuç olarak ortaya çıkan çeyrek daire kesiminde verilmektedir.

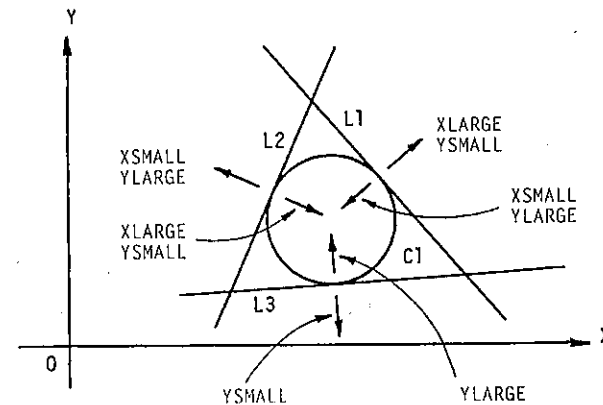
Dolayısıyla, Şekil 4-1'deki P1 noktası için ya XSMALL ya da YSMALL kullanılabilir. Bazen, verilen bir geometrik şekilden önce gelen bir doğrultu değiştirici, tanımlanacak geometrik şeklin verilene göre hangi tarafa yerleştirildiğini göstermek için de kullanılır. Örneğin, verilen üç doğruyla teğet olan bir çemberi tanımlayan deyim şöyledir (Şekil 4-3).

CIRCLE / $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\} .L1, \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\} .L2, \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\} .L3$

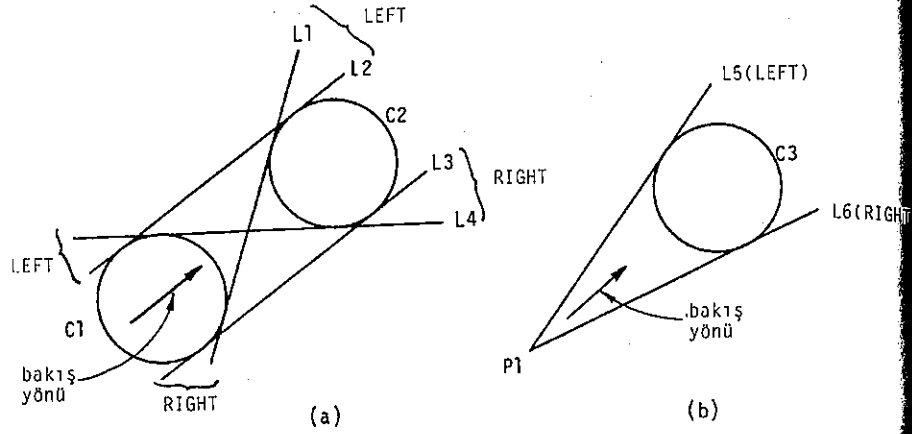
Tanımlanan çemberin sırasıyla L1, L2 ve L3 doğrularına göre konumu göstermek için; L1, L2 ve L3'den önce gelen değiştiriciler seçilebilir. Bir kural olarak, seçilen doğrultu değiştiricinin mümkün olan iki çözüm arasında önemli bir fark gösterebilmesi gerekir. Örneğin Şekil 4-3'te, L3 doğrusundan önce Y yönü için bir değiştirici (YLARGE veya YSMALL) kullanılması önerilir; çünkü her iki taraftaki çözümlerin X koordinatları arasındaki fark önemlidir.

RIGHT (SAĞ) ve LEFT (SOL) ön değiştiricileri, verilen bir daireye ait teğet doğrusunun, referans noktasından (veya çemberden) söz konusu dairenin merkezine doğru bakıldığında dairenin sağ veya sol tarafında olduğunu göstermek için kullanılır (Şekil 4-4).

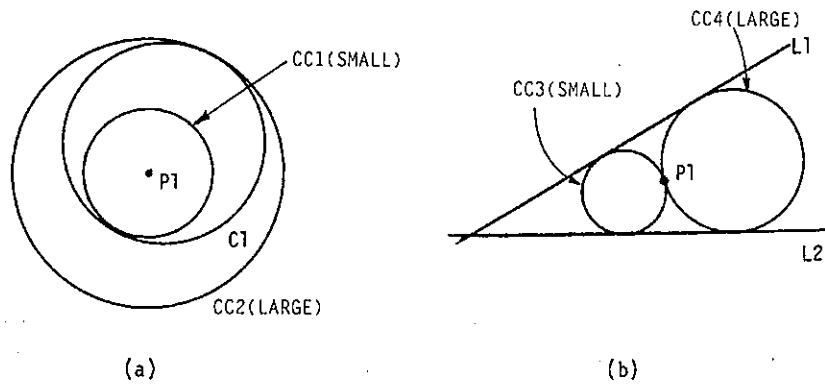
Bir daire yayı uzunluğunun veya açısının belli bir referans noktasına göre ölçüldüğü yönü göstermek için ise, saatin dönme yönü anlamına gelen CLW ve saat dönme yönünün tersi anlamına gelen CCLW doğrultu değiştiricilerden yararlanır.



Şekil 4-3 Tanımlanmış geometrik öğenin (C1 daresi), verilen bir doğrunun hangi tarafında olduğunu göstermek için doğrultu değiştiricinin kullanımı. C1 daresi L1'in XSMALL veya YSMALL tarafındadır. X yönündeki XLARGE ve XSMALL değiştiriciler her iki taraftaki çözümlerin koordinatları arasında önemli bir fark göstermediklerinden L3 doğrusu için tavsiye edilmez.



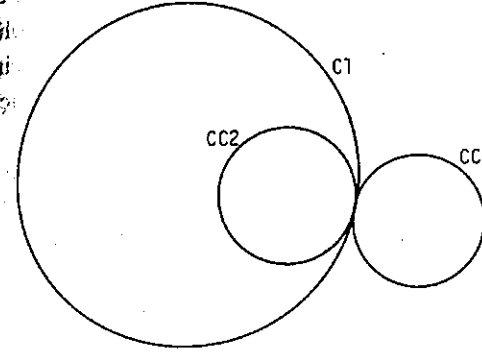
Şekil 4-4 Tanımlanmış olan doğruların verilen çemberin hangi tarafında yerleşik olduğunu göstermek için LEFT (SOL) ve RIGHT (SAĞ) doğrultu değiştiricilerin kullanımı. (a) C1 çemberinin merkezinden C2 çemberininkine doğru bakıldığında, L1 doğrusu C1'in sağ tarafında ve C2'nin de sol tarafındadır. (b) P1 tarafından C3 çemberine doğru bakıldığında, L5 ve L6 doğruları C3 çemberinin sırasıyla solunda ve sağındadır.



Şekil 4-5 LARGE (BÜYÜK) ve SMALL (KÜÇÜK) doğrultu değiştiriciler, istenen bir çözümün boyutunun diğerine göre bağlı değerini gösterir. (a) Verilen bir P1 merkezi ve C1 teğet daireleri için iki çözüm (CC1 ve CC2 daireleri) vardır. İstenen dairenin bağlı boyutuna göre bir doğrultu değiştirici seçilebilir. (b) Verilen iki teğet doğrusu (L1 ve L2) ve istenen dairenin çemberi üzerinde verilen bir P1 noktası için mümkün olan iki çözüm CC3 ve CC4 daireleridir.

Doğrultu değiştiricilerin anlamları açıkça bellidir. LARGE veya SMALL, istenen çözümün mümkün olan diğer çözüme kıyasla daha büyük veya daha küçük ölçüde olduğunu ifade eder (Şekil 4-5).

IN veya OUT ise, söz konusu deyimle tanımlanan geometrik öğenin değiştiriciden sonra belirtilmiş olan geometrik öğeyle ortak bir hacim veya alanı paylaşıp paylaşmadığını gösterir.



Şekil 4-6 IN (İç) ve OUT (Dış) ilişki değiştiriciler CC2 daireleri C1 dairesinin içinde veya C1 dairesinin içinde veya C1 daireleri CC2 dairesinin içindedir; çünkü bunların ortak bir alanı mevcuttur. CC1 daireleri ise C1 ve CC2'nin dışındadır.

Bir değiştiricinin (modifikatörün) kendisinden sonra gelen geometrik öğeyi etkilediğine dikkat edilmesi gerekir. Aşağıdaki diyagram, değiştiriciler ile bunlarla ilgili geometrik öğelerin arasındaki ilişkileri göstermektedir:

C1=CIRCLE/LARGE, YLARGE, TANTO, L1, XSMALL, TANTO, L2, THRU, P1



Bu deyim, verilen bir P1 noktasından geçen bir C1 dairesini tanımlar. Bu daire, verilen teğet doğrusunu (L2) XSMALL tarafından ve L1 teğet doğrusunun da YLARGE tarafından. Ayrıca bu daire, mümkün olan iki teğet dairenin daha büyük olanıdır.

Bazen bir değiştiriciden sonra birden fazla geometrik öğe bulunur (Tablo 4-3'deki No.4 deyim formatına bakınız). Böyle bir durumda değiştirici yalnızca bu geometrik öğelerin sonucunda mümkün olabilecek çözümler arasında doğru olanı göstermek için kullanılır.

Daha sonra görüleceği üzere, APT programındaki geometrik tanım deyimlerinin birçoğu X-Y düzlemindeki geometrik öğeleri tanımlamak için kullanılır. Bunun sebebi, nümerik kontrollü (NC) freze tezgahlarında Z ekseninin daima mil (veya freze kesicisi) eksenine ile çakışması ve çoğu durumlarda çakı hareketinin X-Y düzleminde olmasıdır.

Diğer deyimler içinde başvurulabilmesi için, bir geometrik öğeye bir isim (sembol) verilebilir. Normal olarak, bir sembol tanımlandıktan sonra artık diğer bir geometrik öğenin ismi olarak kullanılmaz. Örneğin eğer P1 noktası:

$$P1=POINT/1,2,3$$

deyimiyle tanımlanmakta ise, P1 noktasının (1,2,3) noktası dışında herhangi bir geometrik öğe için sembol olarak kullanılmasına izin verilemez. Eğer bir sembolün programlama gereği nedeniyle yeniden tanımlanması zorunlu ise, NC işlemcisinin geometrik öğe sembollerinin yeniden tanımlanabilmesine imkan sağlamak için komut veren bir deyim belirtilmesi gerekir (Bölüm 4.11'e bakınız).

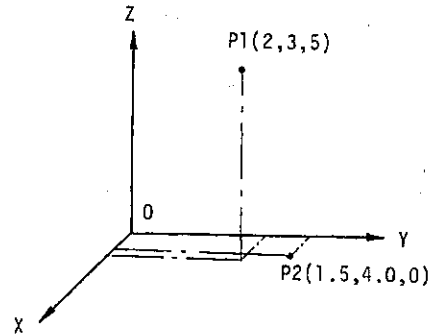
Aşağıda, çeşitli geometrik tanım deyimleri için kullanılabilir formatlar incelenmektedir. Burada incelenmemiş olan ilave formatlar için, okuyucunun IBM APT-AC SD işlemcisi ile ilgili kullanma kılavuzlarına^{1,2} başvurması gerekir.

4.1 BİR NOKTAYI TANIMLAYAN DEYİMLER

APT dilinde, uzaydaki bir noktayı tanımlamak için kullanılabilen 26 deyim formatı vardır. Bunlar şu üç tip altında sınıflandırılabilir: bir noktayı koordinatlarıyla tanımlayanlar; bir noktayı verilen bir geometrik öğe üzerindeki veya söz konusu geometrik öğeye izafi konumu ile tanımlayanlar; ve bir noktayı iki veya daha fazla geometrik öğenin kesişimi olarak tanımlayanlar. Bu gruplarda yer alan deyimler sırasıyla Tablo 4-1, 4-2 ve 4-3'de verilmektedir.

TABLO 4-1 BİR NOKTAYI KOORDİNATLARIYLA TANIMLAYAN DEYİMLER

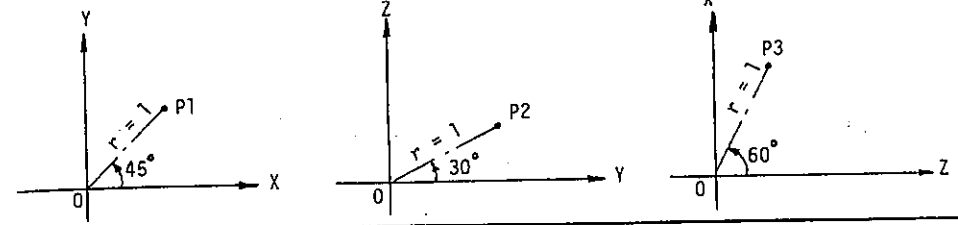
No.	Bilinen Durum	Deyim Formatı
1	Noktanın Kartezyen koordinatları (x,y,z) verilmiştir.	$POINT / \begin{Bmatrix} x,y,z \\ x,y \end{Bmatrix}$



Yorum Eğer ikinci seçenek seçilmişse (yani sadece x ve y koordinatları belirtilmişse), z koordinatı, bu deyimden bir önce gelen ZSURF deyimini (Bölüm 4.10'a bakınız) ile belirlenir ya da, eğer hiçbir ZSURF deyimini belirtilmemiş ise, sıfırdır.

TABLO 4-1 BİR NOKTAYI KOORDİNATLARIYLA TANIMLAYAN DEYİMLER (Devam)

No.	Bilinen Durum	Deyim Formatı
Örnekler		
	P1 = POINT/2,3,5 P2 = POINT/ 1.5, 4.0,0 (veya P2=POINT/1.5, 4.0)	
Burada, P1 ve P2 semboldür		
2	Nokta X-Y, Y-Z düzleminde ve kutupsal koordinatları (yani r kutupsal uzaklığı ve a açısı) ile tanımlanmaktadır.	$POINT/RTHETA, \begin{Bmatrix} XYPLAN \\ XZPLAN \\ ZXPLAN \end{Bmatrix}, r,a$ $POINT/THETAR, \begin{Bmatrix} XYPLAN \\ YZPLAN \\ ZXPLAN \end{Bmatrix}, a,r$



Yorum XYPLAN, YZPLAN ve ZXPLAN kelimeleri, noktanın sırasıyla X-Y, Y-Z ve Z-X düzleminde olduğunu gösterir. RTHETA ve THETAR ise, parametrelerin belirtildiği sırayı göstermektedir.

Örnekler

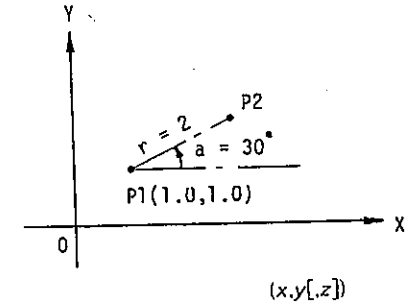
P1 = POINT/RTHETA, XYPLAN, 1,45
P2=POINT/THETAR, YZPLAN, 30, 1
P3=POINT/RTHETA, ZXPLAN, 1,60

3 Nokta X-Y düzleminde ve verilen bir P1 (x,y) noktasına göre kutupsal koordinatlarla (r,x) tanımlanmaktadır.

$$POINT / \begin{Bmatrix} x,y \\ r \end{Bmatrix}, RADIUS, r,a$$

Örnekler

P2=POINT/P1, RADIUS, 2.0,30
veya
P2=POINT/1.0,1.0, RADIUS, 2.0, 3.0

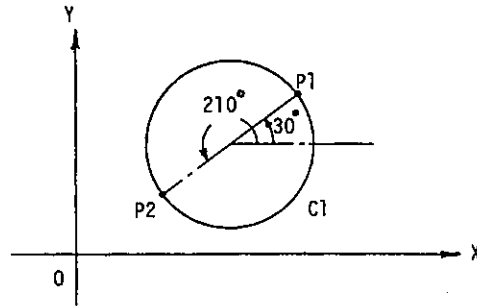


TABLO 4-1 BİR NOKTAYI KOORDİNATLARIYLA TANIMLAYAN ADEYİMLER (Devam)

No.	Bilinen Durum	Deyim Formatı
4	Nokta içiçte Kartezyen koordinatlarıyla (x,y,z) tanımlanmaktadır.	
	Yorum Bu format, önceki deyimlerde tanımlanmamış olan bir nokta sembolünün yerine olmak üzere bir geometrik tanım deyiminde kullanılır. z koordinatı, eğer yoksa, sıfırdır.	
	Örnek Format No.3'te, P1 sembolünün yerine geçmek üzere (1.0, 1.0,0)'ı kullanabiliriz.	
	P2=POINT / (1.0,1.0,0), RADIUS, 2.0,30	

TABLO 4-2 BİR NOKTAYI BİR GEOMETRİK ÖĞE ÜZERİNDEKİ VEYA SÖZ KONUSU ÖĞEYE İZAFİ KONUMUNA GÖRE TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Nokta, X-Y düzleminde verilen bir C1 çemberinin üzerinde olup, bu noktadan geçen yarıçap ile X eksenini arasında a açısı vardır.	POINT/C1, ATANGL, a



Yorum ATANGL, "...'lik açıda" anlamına gelen bir sözcüktür.

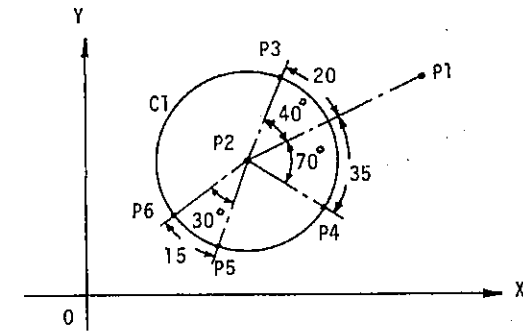
Örnekler

P1=POINT/C1, ATANGL, 30

P2=POINT/C1, ATANGL, 210

TABLO 4-2 BİR NOKTAYI BİR GEOMETRİK ÖĞE ÜZERİNDEKİ VEYA SÖZ KONUSU ÖĞEYE İZAFİ KONUMUNA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
2	Nokta X-Y düzleminde verilen bir C1 çemberi üzerinde olup, verilen bir P1 noktası tarafından belirlenen ve çember üzerindeki referans konumuna göre bir s daire yayı uzaklığındadır.	POINT/P1, DELTA, $\left\{ \begin{matrix} CLW \\ CCLW \end{matrix} \right\}$, ON, C1, ARC, s



Yorum DELTA ve ARC sözcükleri, s uzunluğunun referans noktasından itibaren ölçülen bir artışlı uzaklık olduğunu göstermektedir. Verilen P1 noktası, X-Y düzleminde herhangi bir yere yerleştirilebilir. Referans noktası, P1 ve verilen C1 dairesinin merkezini birleştiren doğrunun çemberi kestiği noktadır. Eğer P1 noktası verilen C1 dairesinin merkezi ise; bu durumda referans noktası, verilen daire ile merkezden geçen ve +X yönünde uzatılan yatay doğrunun kesişimidir.

Örnek

P3=POINT/P1, DELTA, CCLW, ON, C1, ARC, 20

P4= POINT/P1, DELTA, CLW, ON, C1, ARC, 35

P6=POINT/P5, DELTA, CLW, ON, C1, ARC, 15

3	Nokta, verilen bir C1 daire çemberi üzerindedir ve verilen bir P1 noktasından a açısı uzaklıktadır.	POINT/P1, DELTA, $\left\{ \begin{matrix} CLW \\ CCLW \end{matrix} \right\}$, ON, C1, ATANGL, a
---	---	---

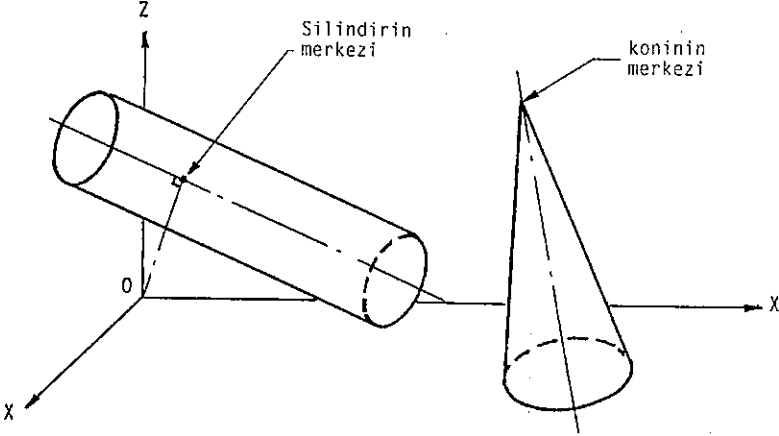
Yorum Bu deyim bir öncekine benzerdir. Fakat burada bir yay uzunluğu yerine bir açı değeri kullanılmıştır.

Örnek (No.2 Deyim formatına bakınız).

P3=POINT/P1, DELTA, CCLW, ON, C1, ATANGL, 40

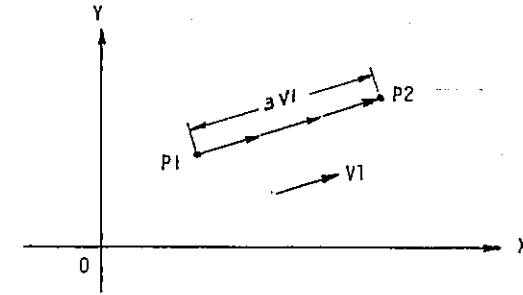
P6=POINT/P5, DELTA, CLW, ON, C1, ATANGL, 30

TABLO 4-2 BİR NOKTAYI BİR GEOMETRİK ÖĞE ÜZERİNDEKİ VEYA SÖZ KONUSU ÖĞEYE İZAFİ KONUMUNA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
4	Nokta; verilen bir C1 dairesinin, küresinin, yarım küresinin, konisinin veya silindirin merkezi.	POINT/CENTER, C1
		
<p>Yorum Bir koninin merkezi tepe noktasıdır. APT dilinde bir silindirin merkezi, önceki bir CYLNDR deyiminde silindirin merkezi olarak tanımlanmış olan noktadır. Eğer merkez tanımlanmamış ise, silindir eksenine koordinat sisteminin başlangıç noktasından eksene çizilen normalin kesişimi verilen silindirin merkezidir.</p>		
5	Nokta, verilen bir P1 noktasından itibaren n.V1 vektörel yerdeğişim ile tanımlanır.	POINT/P1, DELTA, n TIMES, [UNIT,]V1

TABLO 4-2 BİR NOKTAYI BİR GEOMETRİK ÖĞE ÜZERİNDEKİ VEYA SÖZ KONUSU ÖĞEYE İZAFİ KONUMUNA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
<p>Yorum Eğer isteğe bağlı UNIT (Birim) sözcüğü kullanılırsa, V1 vektörü bir birim vektör olarak dikkate alınır. Yani, bu durumda $V1 = 1$.</p>		



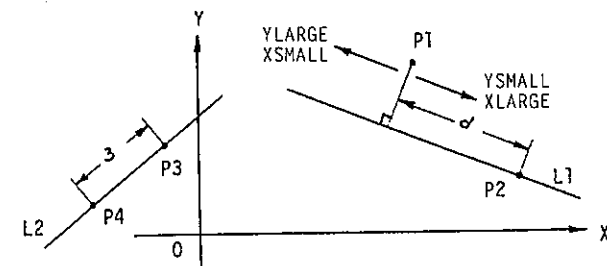
Örnek

P2=POINT/P1, DELTA, 3, TIMES, V1

- 6 Nokta X-Y düzlemi içinde bir L1 doğrusu üzerindedir ve verilen bir P1 noktasından d uzaklıktadır.

POINT/ $\begin{cases} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{cases}$, ON, L1, DELTA, d, P1

Yorum Verilen P1 noktası X-Y düzlemi içinde herhangi bir yerde bulunabilir. Eğer bu nokta L1 doğrusu üzerinde değilse; "d" uzaklığının ölçüleceği referans noktasını belirlemek için, verilen L1 doğrusuna dik bir izdüşüm doğrusu çizilir.



Örnek

P2=POINT/YSMALL, ON, L1, DELTA, d, P1

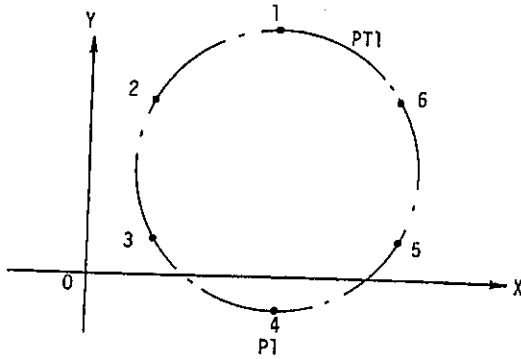
P4=POINT/XSMALL, ON, L2, DELTA, 3, P3

TABLO 4-2 BİR NOKTAYI BİR GEOMETRİK ÖĞE ÜZERİNDEKİ VEYA SÖZ KONUSU ÖĞEYE İZAFİ KONUMUNA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
7	Nokta, bir PT1 paterinin (modelinin) "n"nci noktasıdır.	POINT/PT1, n

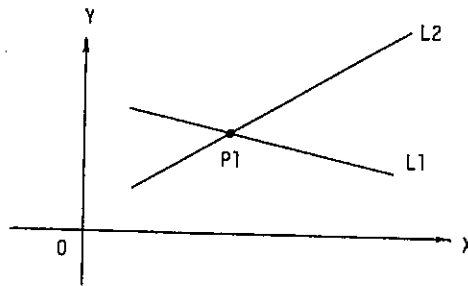
Örnek

P1=POINT/PT1, 4 \$\$\$PT1 ALTI NOKTADAN OLUŞAN BİR MODELDIR.



TABLO 4-3 BİR NOKTAYI İKİ VEYA DAHA FAZLA SAYIDA GEOMETRİK ÖĞENİN KESİŞİMİ OLARAK TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Nokta verilen L1 ve L2 doğrularının kesişimidir.	POINT/INTOF, L1, L2



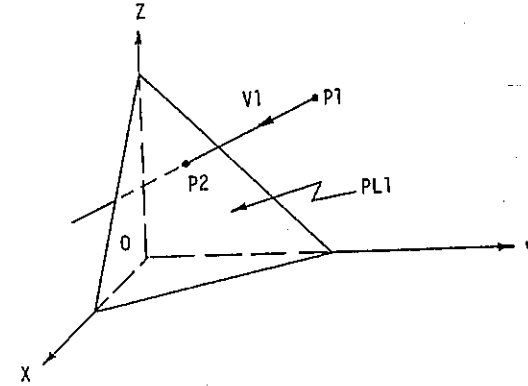
Örnek

P1=POINT/INTOF, L1, L2

*Bölüm 5'de PATTERN tanımlamasına bakınız.

TABLO 4-3 BİR NOKTAYI İKİ VEYA DAHA FAZLA SAYIDA GEOMETRİK ÖĞENİN KESİŞİMİ OLARAK TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
2	Nokta, bir PL1 düzlemi ile bir V1 noktasından geçen bir V1 vektörünün kesişimi olarak tanımlanır.	POINT/INTOF, PL1, V1, P1

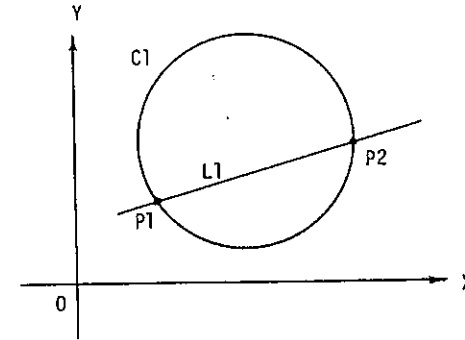


Örnek

P2=POINT/INTOF, PL1, V1, P1

- 3 Nokta X-Y düzlemi içinde olup, verilen bir L1 doğrusu ile verilen bir C1 dairesinin kesişimidir.

POINT/ $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}$ /INTOF, L1, C1



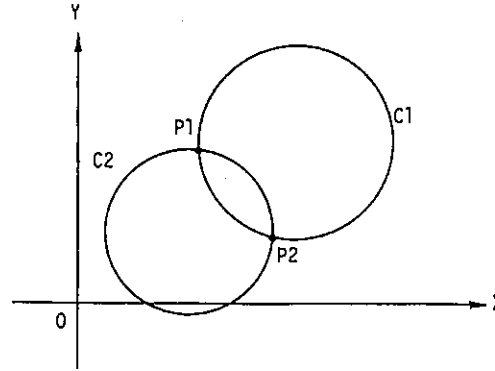
Yorum Genel olarak iki çözüm vardır. İstlenen çözümü göstermek için yön değiştirici seçilir.

TABLO 4-3 BİR NOKTAYI İKİ VEYA DAHA FAZLA SAYIDA GEOMETRİK ÖĞENİN KESİŞİMİ OLARAK TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
Örnek		
	P1=POINT/XSMALL, INTOF, L1, C1	
	P2=POINT/YLARGE, INTOF, L1, C1	

- 4 Nokta verilen iki dairenin (C1 ve C2) kesişimidir.

POINT/ $\begin{cases} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{cases}$,INTOF,C1,C2

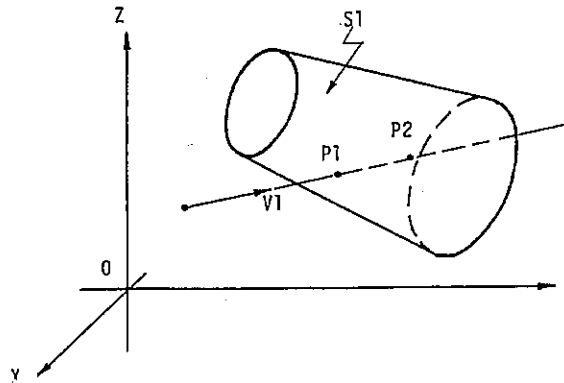


Örnek

P1=POINT/XSMALL, INTOF, C1, C2
P2=POINT/YSMALL, INTOF, C1, C2

- 5 Nokta, verilen bir V1 nokta vektörünün (veya uzayda alınan bir doğrunun) bir APT yüzeyi (S1) ile kesişimidir.

POINT/n, INTOF, V1, S1



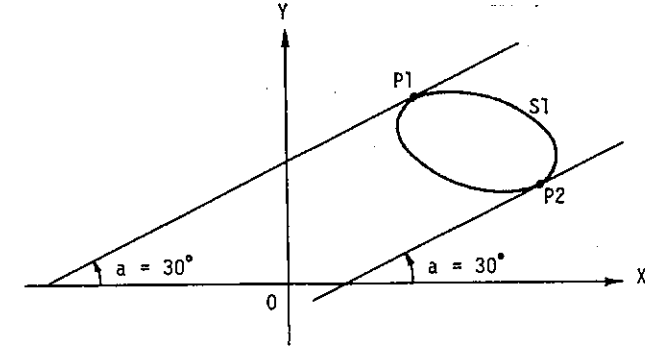
Yorum Bir nokta vektörü (ya da uzayda alınan bir doğru), uzaydaki sabit bir noktadan geçen bir vektördür (Bölüm 4.4'e bakınız). S1 ise şu yüzeylerden biri olabilir: Daire, koni, silindir, elips, hiperbol, GCONIC, LCONIC (Bölüm 4.5'e bakınız), düzlem, çeyrek yüzey veya küre.

TABLO 4-3 BİR NOKTAYI İKİ VEYA DAHA FAZLA SAYIDA GEOMETRİK ÖĞENİN KESİŞİMİ OLARAK TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
Örnek		
	P2=POINT/2, INTOF, V1, S1	

- 6 Nokta, X-Y düzlemindeki bir S1 eğri yüzeye X eksenine ile belirli bir a açısı yapan teğetin değme noktasıdır.

POINT/ $\begin{cases} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{cases}$,ON,S1,ATANGL,a



Yorum S1 yüzeyi şu yüzeylerden biri olabilir: daire, elips, hiperbol, GCONIC ve LCONIC.

Örnek

P1=POINT/YLARGE, ON, S1, ATANGL, 30
P2=POINT/XLARGE, ON, S1, ATANGL, 30

4.2 BİR DOĞRUYU TANIMLAYAN DEYİMLER

Bir doğru

LINE/.....

deyimi ile tanımlanabilir.

APT dilinde, LINE (Doğru) deyimi için 27 format vardır. Bu formatlar şu üç gruba ayrılabilir:

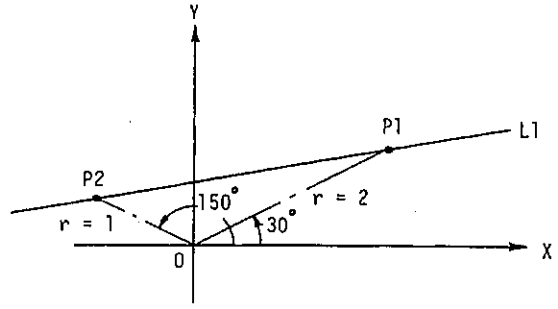
- Verilen iki noktaya göre ya da verilen bir nokta ile ilişkisine göre veya her iki durum tarafından tanımlanan bir doğru.
- Bir dairenin veya dairelerin referans olarak kullanılmasıyla tanımlanan bir doğru.
- Üzerindeki bir nokta ile doğrunun elips, konik ve TABCYL (tablolaştırılmış silindir) gibi bir APT geometrik öğesiyle olan ilişkisine göre tanımlanan bir doğru.*

Bu üç deyim grubunun listesi, sırasıyla Tablo 4-4, 4-5 ve 4-6'da gösterilmektedir.

* Bir konik ve tablolaştırılmış silindirin tanımları için, Bölüm 4.5 ve 4.7'ye bakınız.

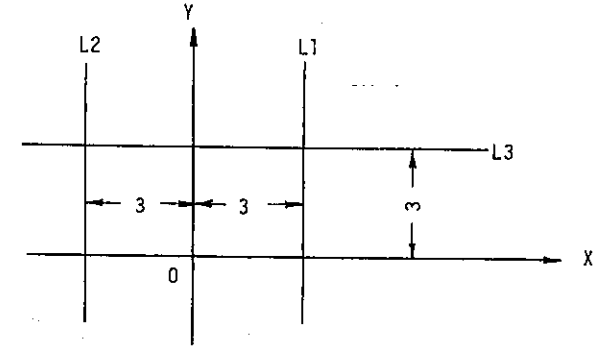
APT programlama kılavuzlarına ^{1,2} göre, LINE deyimiyle tanımlanan bir doğru, bir X-Y düzleminin bu düzleme dik olan bir düzlem ile kesişimidir.

TABLO 4-4 BİR DOĞRUYU VERİLEN İKİ NOKTA İLE, VEYA VERİLEN BİR NOKTA VE VERİLEN BİR DOĞRU YA DA DOĞRULAR İLE, VEYA HER İKİ DURUMA GÖRE TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Doğru, verilen iki noktadan P1 (x1, y1, z1) ve P2 (x2, y2, z2) geçmektedir.	LINE/ $\left\{ \begin{array}{l} x1, y1, z1, x2, y2, z2 \\ x1, y1, x2, y2 \\ P1, P2 \end{array} \right\}$
	Yorum NC işlemcisi içinde yukarıda sırasıyla belirtilen formatların birinci ve ikinci seçenekleri tarafından tanımlanan doğrular aynıdır. Üçüncü seçenek seçildiğinde ise, P1 ve P2 sembollerinin daha önce tanımlanmış olmaları gerekir.	
2	Doğru, X-Y düzleminde bulunan ve sırasıyla (r1, a1) ve (r2, a2) kutupsal koordinatlarıyla tanımlanan iki noktadan geçmektedir.	LINE/ $\left\{ \begin{array}{l} RTHETA, r1, a1, r2, a2 \\ THETA, a1, r1, a2, r2 \end{array} \right\}$
		
	Örnekler L1=LINE/RTHETA, 2,30,1,150 L2=LINE/THETA, 30,2,150,1	
3	Doğru, X veya Y eksenine paralel ve söz konusu eksenin bir d mesafesindedir.	LINE/ $\left\{ \begin{array}{l} XAXIS \\ YAXIS \end{array} \right\} ,d$

TABLO 4-4 BİR DOĞRUYU VERİLEN İKİ NOKTA İLE, VEYA VERİLEN BİR NOKTA VE VERİLEN BİR DOĞRU YA DA DOĞRULAR İLE, VEYA HER İKİ DURUMA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum XAXIS ve YAXIS sözcükleri, sırasıyla X ve Y eksenlerinin referans doğrusu olarak kullanıldığını belirtir. d uzaklığın işareti ise, tanımlanan doğrudan referans eksenine göre bulunduğu tarafı gösterir.	

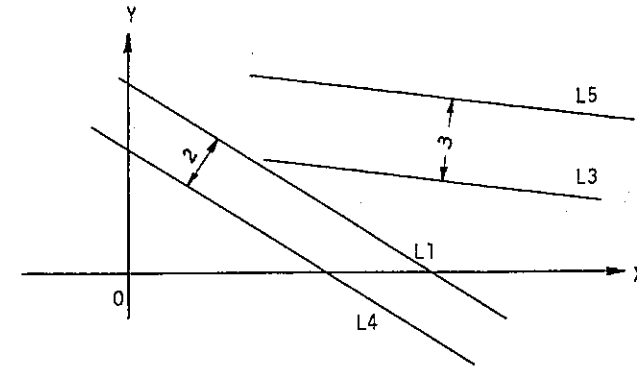


Örnekler

L1=LINE/YAXIS, 3,0
L2=LINE/YAXIS, -3,0
L3=LINE/XAXIS, 3,0

- 4 Doğru, verilen bir L1 doğrusuna paralel ve bu doğrudan bir d mesafesindedir.

LINE/PARLEL, L1, $\left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \end{array} \right\} ,d$

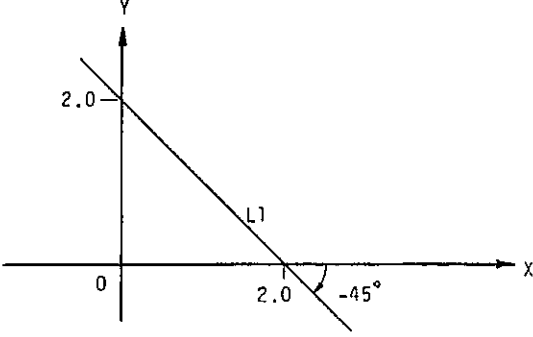


Yorum PARLEL kelimesi "... e paralel " anlamına gelir.

Örnekler

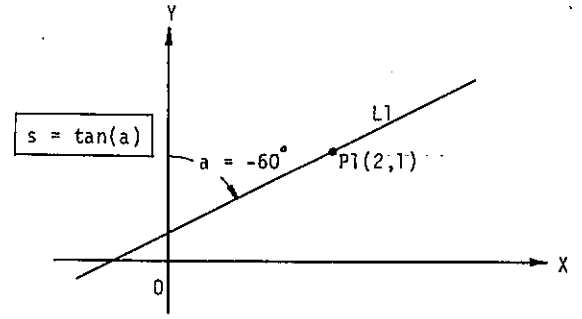
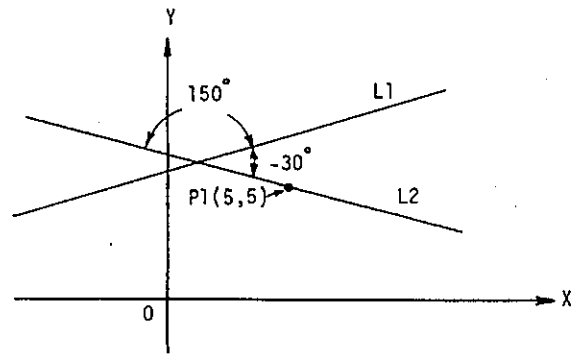
L4=LINE/PARLEL, L1, XSMALL, 2
L5=LINE/PARLEL, L3, YLARGE, 3

TABLO 4-4 BİR DOĞRUYU VERİLEN İKİ NOKTA İLE, VEYA VERİLEN BİR NOKTA VE VERİLEN BİR DOĞRU YA DA DOĞRULAR İLE, VEYA HER İKİ DURUMA GÖRE TANIMLAYAN DEYİMLER (devam)

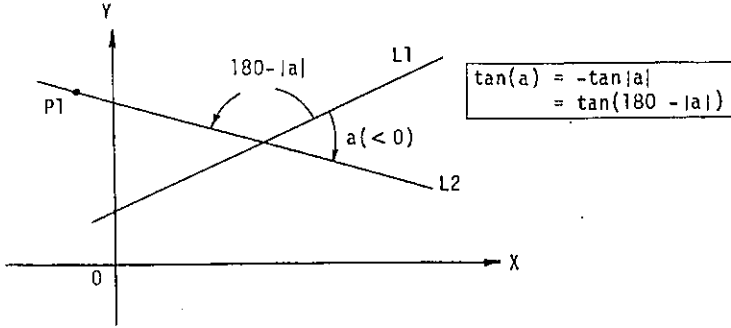
No.	Bilinen Durum	Deyim Formatı
5	Doğru, +X eksenine göre bir s * eğimine ve X veya Y eksenini üzerinde bir d kesişim mesafesine sahiptir.	LINE/SLOPE, s , INTERC, $\left\{ \begin{array}{l} \text{XAXIS} \\ \text{YAXIS} \end{array} \right\} d$
		
	Yorum INTERC sözcüğü, "kesişim mesafesi" anlamına gelir.	
	Örnekler L1=LINE/SLOPE, -1, INTERC, 2 L2=LINE/SLOPE, -1, INTERC, XAXIS, 2	
6	Doğru, +X eksenini ile a açısı yapmakta ve x veya y eksenini bir d mesafesinde kesmektedir.	LINE/ATANGL, a , /INTERC, $\left\{ \begin{array}{l} \text{XAXIS} \\ \text{YAXIS} \end{array} \right\} d$
	Örnekler (No.5 deyim formatındaki şekle bakınız) L1=LINE/ATANGL, -45, INTERC, 2 L1=LINE/ATANGL, 135, INTERC, 2 L1=LINE/ATANGL, -45, INTERC, XAXIS, 2 L1=LINE/ATANGL, 135, INTERC, XAXIS, 2	
7	Doğru, verilen bir P1 (x,y) noktasından geçmekte ve +X veya +Y eksenini ile bir a açısı yapmaktadır.	LINE/ $\left\{ \begin{array}{l} \text{XY} \\ \text{P1} \end{array} \right\}$, ATANGL, a , $\left\{ \begin{array}{l} \text{XAXIS} \\ \text{YAXIS} \end{array} \right\}$

*1. ve 2. referansta, bir açısal değer bu yazılımdaki "s" için kullanılır. Bu kullanım yanlıştır.

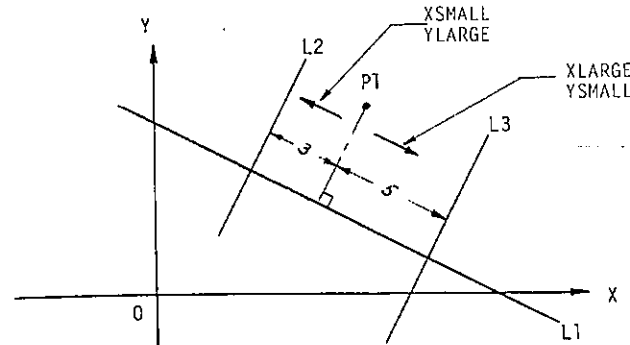
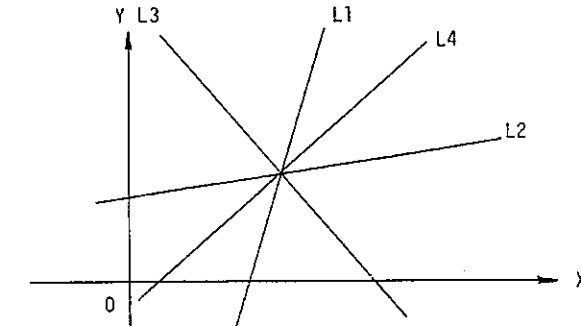
TABLO 4-4 BİR DOĞRUYU VERİLEN İKİ NOKTA İLE, VEYA VERİLEN BİR NOKTA VE VERİLEN BİR DOĞRU YA DA DOĞRULAR İLE, VEYA HER İKİ DURUMA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
		
	Örnekler L1=LINE/2, 1, ATANGL, -60, YAXIS L1=LINE/2,1, ATANGL, 30 L1=LINE/P1, ATANGL, 120, YAXIS	
8	Doğru, bir P1 noktasından geçmekte olup; +Y eksenine göre bir s eğimine sahiptir.	LINE/P1, SLOPE, s $\left\{ \begin{array}{l} \text{XAXIS} \\ \text{YAXIS} \end{array} \right\}$
	Örnekler (No.7 deyim formatındaki şekle bakınız) L1=LINE/P1, SLOPE, -1.7321, YAXIS veya L1=LINE/P1, SLOPE, 0.5774	
9	Doğru, verilen bir P1 (x,y) noktasından geçmekte ve verilen bir L1 doğrusu ile bir a açısı yapmaktadır.	LINE/ $\left\{ \begin{array}{l} \text{XY} \\ \text{P1} \end{array} \right\}$, ATANGL, a ,L1
	Yorum Aç, pozitif değer saata yönünün tersinde olacak şekilde, verilen doğruya ölçülür.	
		

TABLO 4-4 BİR DOĞRUYU VERİLEN İKİ NOKTA İLE, VEYA VERİLEN BİR NOKTA VE VERİLEN BİR DOĞRU YA DA DOĞRULAR İLE, VEYA HER İKİ DURUMA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Örnek L2=LINE/P1, ATANGL, -30,L1 veya L2=LINE/5.5, ATANGL, 150, L1	
10	Doğru, verilen bir P1 noktasından geçmekte olup; verilen bir L1 doğrusuna göre bir s eğimine sahiptir.	LINE/P1, SLOPE, s,L1
		
	Örnek Eğer a=-60 derece ise, şu format yazılabilir: L2=LINE/P1, SLOPE, -1, 732,L1	
11	Doğru, verilen bir P1 (x,y) noktasından geçmekte olup; verilen bir L1 doğrusuna paraleldir.	LINE / $\begin{Bmatrix} xy \\ P1 \end{Bmatrix}$, PARLEL,L1
12	Doğru, verilen bir P1 (x,y) noktasından geçmekte olup; verilen bir L1 doğrusuna diktir (PERPTO).	LINE / $\begin{Bmatrix} xy \\ P1 \end{Bmatrix}$, PERPTO,L1
13	Doğru, verilen bir L1 doğrusuna diktir ve verilen bir P1 noktasından d uzaklıktadır.	LINE / $\begin{Bmatrix} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \end{Bmatrix}$, PERPTO, L1, DELTA, d, P1

TABLO 4-4 BİR DOĞRUYU VERİLEN İKİ NOKTA İLE, VEYA VERİLEN BİR NOKTA VE VERİLEN BİR DOĞRU YA DA DOĞRULAR İLE, VEYA HER İKİ DURUMA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
		
	Yorum İstenen doğrunun verilen P1 noktasının hangi tarafında olduğunu belirtmek için doğrultu değiştiriciler kullanılır.	
	Örnekler L2=LINE/XSMALL, PERPTO, L1, DELTA, 3,P1 L2=LINE/YLARGE, PERPTO, L1, DELTA, 3,P1 L3=LINE/XLARGE, PERPTO, L1, DELTA, 5,P1 L3=LINE/YSMALL, PERPTO, L1, DELTA, 5,P1	
14	Doğru, verilen iki doğru (L1 ve L2) arasındaki açının açı ortayıdır (BISECT).	LINE / $\begin{Bmatrix} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \end{Bmatrix}$, BISECT,L1,L2
	Yorum Verilen L1 ve L2 doğrularını birbirine paralel olması şartıyla, genelde iki çözüm vardır. İstenen çözümün X veya Y koordinat ekseninde daha mı büyük yoksa daha mı küçük kesişim mesafesine sahip olan çözümü göstermek üzere uygun doğrultu değiştirici (XLARGE veya XSMALL, ya da YLARGE veya YSMALL) seçilebilir.	
		

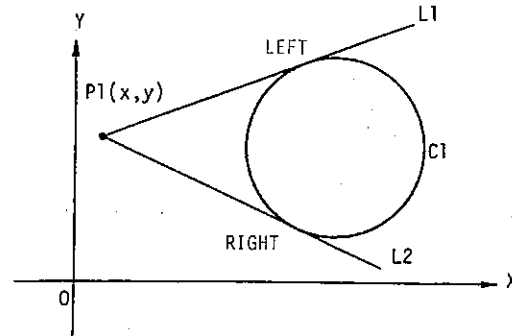
TABLO 4-4 BİR DOĞRUYU VERİLEN İKİ NOKTA İLE, VEYA VERİLEN BİR NOKTA VE VERİLEN BİR DOĞRU YA DA DOĞRULAR İLE, VEYA HER İKİ DURUMA GÖRE TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
Örnekler		
	L3=LINE/XLARGE, BISECT, L1, L2	
	L3=LINE/YLARGE, BISECT, L1, L2	
	L4=LINE/XSMALL, BISECT, L1, L2	
	L4=LINE/YSMALL, BISECT, L1, L2	

TABLO 4-5 BİR DOĞRUYU TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİREYİ VEYA DAİRELERİ KULLANAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Verilen bir P1 (x,y) noktasından geçen doğru, verilen bir C1 dairesine teğettir (TANTO).	LINE/ $\left\{ \begin{array}{l} X \\ Y \\ P1 \end{array} \right\}$, $\left\{ \begin{array}{l} RIGHT \\ LEFT \end{array} \right\}$, TANTO, C1

Yorum LEFT (SOL) veya RIGHT (SAĞ) doğrultu değiştirici, daireye P1 noktasından bakıldığında doğrunun C1 dairesinin solunda veya sağında olduğunu gösterir.



Örnekler

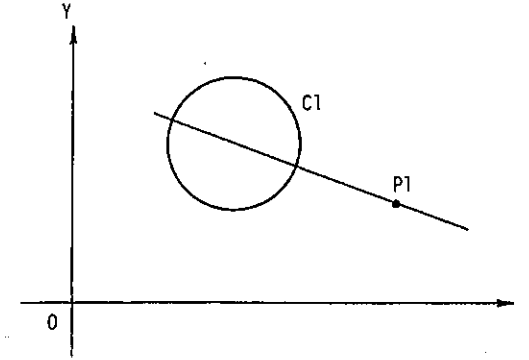
L1=LINE/1.2,2.5, LEFT, TANTO, C1, \$\$X=1.2, Y=2.5

L2=LINE/P1, RIGHT, TANTO, C1

2	Doğru, verilen bir P1 noktasından ve verilen bir C1 dairesinin merkezinden geçmekte olup; dolayısıyla verilen C1 dairesine diktir.	LINE/P1, PERPTO, C1
---	--	---------------------

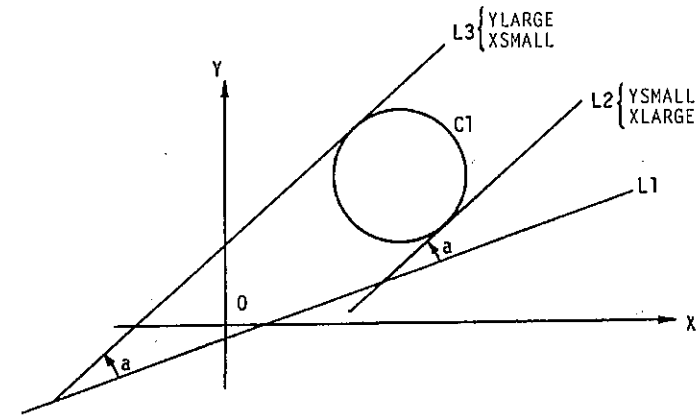
TABLO 4-5 BİR DOĞRUYU TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİREYİ VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
-----	---------------	---------------



- 3 Doğru, verilen bir C1 dairesine teğet olup, verilen bir L1 doğrusu ile bir α açısı yapmaktadır.

LINE/ATANGL, α , L1, TANTO, C1, $\left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \end{array} \right\}$



Yorum α açısı, L1 doğrusundan tanımlanacak olan doğruya doğru ölçülür. Doğrultu değiştirici, iki doğrunun daire ile meydana getirdiği teğet değme noktalarının koordinatlarını karşılaştırmak suretiyle belirlenebilir.

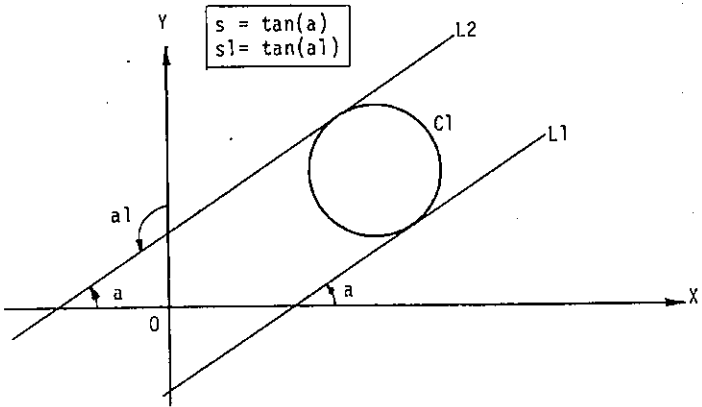
Örnekler

L2=LINE/ATANGL, α , L1, TANTO, C1, XLARGE

L3=LINE/ATANGL, α , L1, TANTO, C1, XSMALL

- 4 Doğru, verilen bir C1 dairesine teğet olup, X veya Y eksenine göre bir α eğimine sahiptir.

TABLO 4-5 BİR DOĞRUYU TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİREYİ VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
		

Örnekler

L1=LINE/XLARGE, TANTO, C1, SLOPE, s
L2=LINE / YLARGE, TANTO, C1, SLOPE, s1, YAXIS

- 5 Doğru, verilen bir C1 dairesine teğet olup; x eksenini ile bir a açısı meydana getirmektedir.

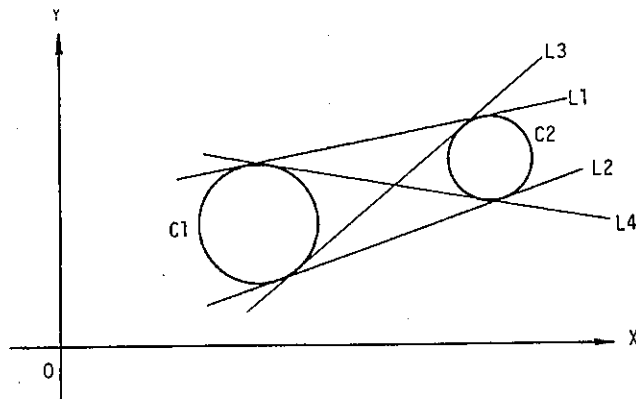
LINE / $\left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \end{array} \right\}$, TANTO, C1, ATANGL, a

Örnekler (No.4 deyim formatındaki şekle bakınız)

L1=LINE/YSMALL, TANTO, C1, ATANGL, a
L2=LINE/XSMALL, TANTO, ATANGL, a

- 6 Doğru, C1 ve C2 dairelerinin ortak teğetidir.

LINE / $\left\{ \begin{array}{l} LEFT \\ RIGHT \end{array} \right\}$, TANTO, C1, $\left\{ \begin{array}{l} LEFT \\ RIGHT \end{array} \right\}$, TANTO, C2

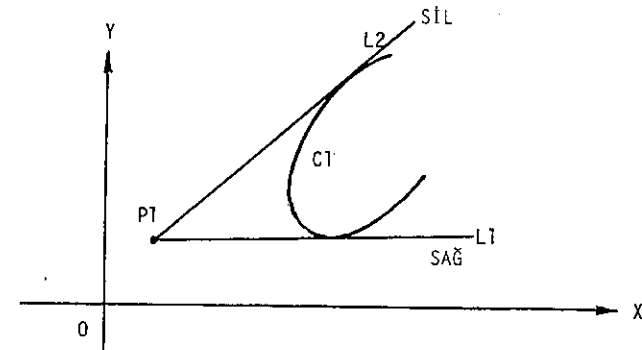


TABLO 4-5 BİR DOĞRUYU TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİREYİ VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum LEFT (SOL) veya RIGHT (SAĞ) doğrultu değiştiricilerden uygun olanının seçilmesi için referans noktası, deyimde belirtilen birinci dairesinin (C1) merkezidir. Dolayısıyla, referans noktasından C2 dairesine doğru bakıldığında; L1 doğrusu C1 ve C2 dairelerinin sol tarafında; L4 doğrusu da C1 dairesinin solunda ve C2 dairesinin sağındadır.	
	Örnekler L2=LINE/RIGHT, TANTO, C1, RIGHT, TANTO, C2 L3=LINE/RIGHT, TANTO, C1, LEF, TANTO, C2	

TABLO 4-6 VERİLEN BİR NOKTADAN GEÇEN VE ÖZEL BİR APT YÜZEYİ İLE BELİRLİ BİR İLİŞKİYE SAHİP BİR DOĞRUYU TANIMLAYAN SEÇİLMİŞ DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Doğru, verilen bir P1 noktasından geçmekte olup; X-Y düzlemi içinde verilen bir eğrinin (C1) merkezidir. Dolayısıyla, referans noktasından C2 dairesine doğru bakıldığında; L1 doğrusu C1 ve C2 dairelerinin sol tarafında; L4 doğru da C1 dairesinin solunda ve C2 dairesinin sağındadır.	LINE/P1, $\left\{ \begin{array}{l} LEFT \\ RIGHT \end{array} \right\}$, TANTO, C1



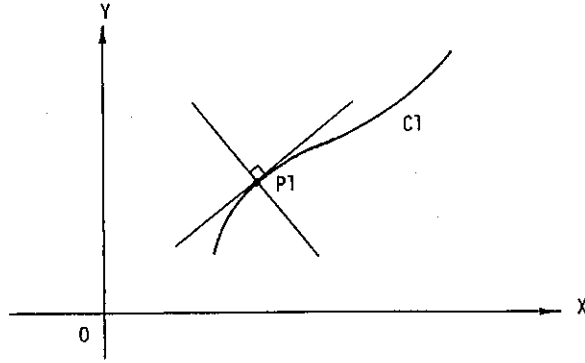
Yorum P1 noktasının verilen C1 eğrisi üzerinde olmaması gerekir.

Örnekler

L1=LINE/P1, RIGHT, TANTO, C1
L2=LINE/P1, LEFT, TANTO, C1

TABLO 4-6 VERİLEN BİR NOKTADAN GEÇEN VE ÖZEL APT YÜZEYİ İLE BELİRLİ BİR İLİŞKİYE SAHİP BİR DOĞRUYU TANIMLAYAN SEÇİLMİŞ DEYİMLER (devam)

No.	Bilinen Şart	Deyim Formatı
2	Doğru, verilen bir C1 eğrisine P1 noktasından teğettir. C1 eğrisi bir elips, hiperbol, LCONIC veya GCONIC olabilir.	LINE/P1,TANTO,C1
3	Doğru, verilen bir C1 eğrisine, yine aynı eğri üzerinde verilen bir P1 noktasında diktir. C1 bir elips, hiperbol, LCONIC veya GCONIC olabilir (No.2 deyim formatındaki şekle bakınız).	LINE/P1,PERPTO,C1



Bu ifade, bir doğrunun nasıl tanımlanmış olduğuna bakılmaksızın, daima X-Y düzlemi içinde olduğu anlamını taşır. Bu, NC işlemcisiyi kullanmak suretiyle, tanımlanmış doğru üzerinde bir noktanın yerini tayin etmeye çalışıldığında doğrudur. Örneğin, NC işlemcisi tarafından bulunan iki doğrunun kesişim noktası, bu iki doğru (Tablo 4-4'deki No.1 deyim formatına göre iki takım nokta koordinatları vasıtasıyla) X-Y düzlemi dışında tanımlanmış olsa veya birbirini kesmese bile, daima X-Y düzleminindedir. Bununla beraber, genelede NC işlemcisi tarafından bir doğru, söz konusu doğruyu içeren ve X-Y düzlemine dik olan bir düzlem olarak dikkate alınır. Bu da, freze kesicisi profil hareketinin tanımlanması için yararlıdır.

4.3 BİR DAİREYİ TANIMLAYAN DEYİMLER

Daire, bir düzlem içinde hareket eden ve sabit bir noktadan uzaklığını sürekli olarak koruyan bir noktanın geometrik yeridir. APT dilinde, bir daire:

CIRCLE/.....

deyimiyle (bölü çizgisinden sonra kelimesi(ler) ve parametre(ler) gelir) tanımlanır. NC işlemcisinde ise bir daire, eksen X-Y düzlemine dik olan sonsuz uzunluktaki bir silindir olarak tanımlanır. Daireyi tanımlayan deyimde belirtilmedikçe, daire merkezinin z koordinatı sıfır kabul edilir.

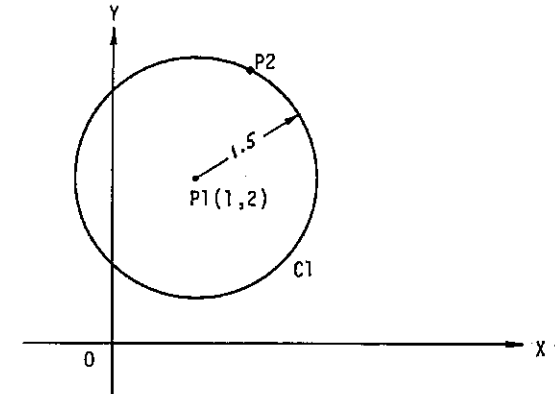
NC işlemcide, CIRCLE (DAİRE) deyimine ait 27 format mevcut olup; bu formatlar şu dört gruba ayrılabilir:

- Bir daireyi tanımlamak için referans olarak bir noktayı veya noktaları kullanan formatlar.
- Bir daireyi tanımlamak için referans olarak noktaları ve doğruları kullanan formatlar.
- Bir daireyi tanımlamak için referans olarak doğruları kullanan formatlar.
- Bir daireyi tanımlamak için referans olarak bir daireyi veya daireleri kullanan formatlar.

Bu deyim formatları, ve her bir deyim grubuna ait örnekler, 4-7'den 4-10'a kadar olan tablolarda verilmiştir.

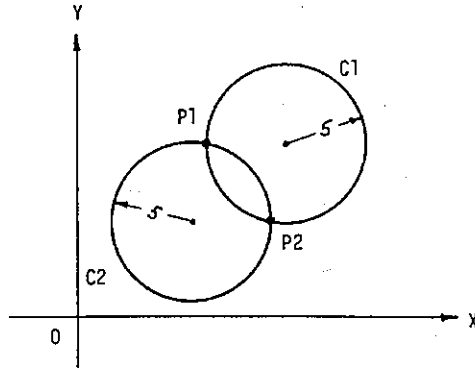
TABLO 4-7 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR NOKTAYI VEYA NOKTALARI KULLANAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Dairenin merkezi P1(x,y) ve yarıçapı r'dir.	CIRCLE/CENTER, $\begin{Bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{Bmatrix}$, RADIUS, r



TABLO 4-7 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR NOKTAYI VEYA NOKTALARI KULLANAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
	Yorum Merkez z koordinatı belirtilmemişse, sıfır olarak alınır.	
	Örnek C1=CIRCLE/CENTER,P1,RADIUS,1.5 veya C1=CIRCLE/CENTER,1,2,RADIUS,1.5	
2	Daire, merkezi P1 noktasında olup; verilen bir P2 noktasından geçmektedir.	CIRCLE/CENTER,P1,P2
	Yorum P1 merkez noktasının, CENTER (MERKEZ) sözcüğünden hemen sonra belirtilmesi gerekir.	
	Örnek (No.1 deyim formatındaki şekle bakınız) C1=CIRCLE/CENTER,P1,P2	
3	Daire; verilen iki noktadan (P1 ve P2) geçmekte olup, verilen bir r yarıçapına sahiptir	CIRCLE/ $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}$, P1, P2, RADIUS, r

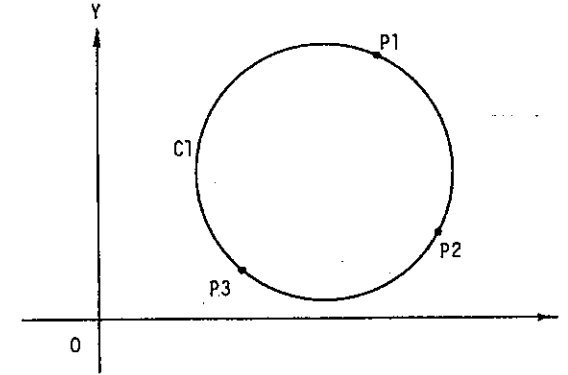


Yorum Mümkün olan iki çözümün merkez koordinatlarını karşılaştırarak, bir doğrultu değiştirici seçilebilir.

Örnek
 C1=CIRCLE/XLARGE,P1,P2,RADIUS,5.0
 veya
 C1=CIRCLE/YLARGE,P1,P2,RADIUS,5.0

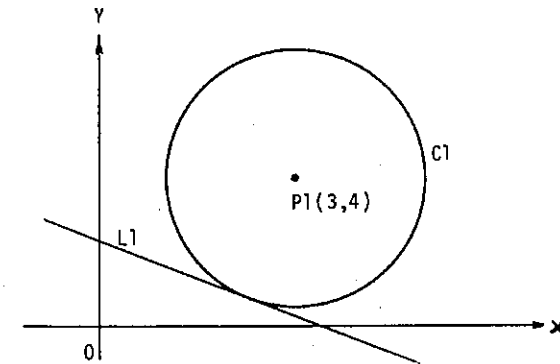
TABLO 4-7 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR NOKTAYI VEYA NOKTALARI KULLANAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
4	Daire, verilen üç noktadan (P1, P2 ve P3) geçmektedir.	CIRCLE/P1,P2,P3



TABLO 4-8 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK NOKTALARI VE DOĞRULARI KULLANAN DEYİMLER

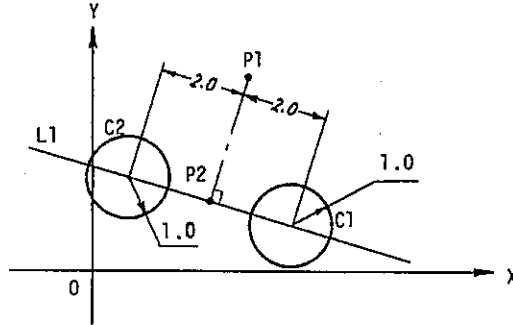
No.	Bilinen Durum	Deyim Formatı
1	Daire; verilen bir L1 doğrusuna teğet (TANTO) olup, merkezi verilen bir P1(x,y) noktasındadır.	CIRCLE/CENTER, $\left\{ \begin{array}{l} X \\ Y \end{array} \right\}$, TANTO, L1



Örnek
 C1=CIRCLE/CENTER,3,4,TANTO,L1
 veya
 C1=CIRCLE/CENTER,P1,TANTO,L1

TABLO 4-8 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK NOKTALARI VE DOĞRULARI KULLANAN DEYİMLER (devam)

No.	Bilinen Şart	Deyim Formatı
2	Dairenin merkezi verilen bir L1 doğrusu üzerinde olup, verilen bir P1 noktasında L1 doğrusu yönünde ölçülen bir d uzaklıktadır. Dairenin yarıçapı ise r 'dir.	CIRCLE/ $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}$, ON,L1,DELTA, d,P1,RADIUS,r



Örnekler

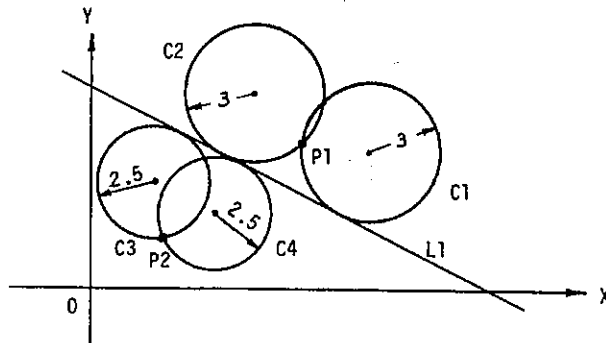
C1=CIRCLE/XLARGE,ON,L1,DELTA,2.0,P1,RADIUS,1.0

C2=CIRCLE/YLARGE,ON,L1,DELTA,2.0,P2,RADIUS,1.0

- 3 Daire; verilen bir L1 doğrusuna teğet olup, verilen bir r yarıçapında, verilen bir P1 noktasından geçmektedir.

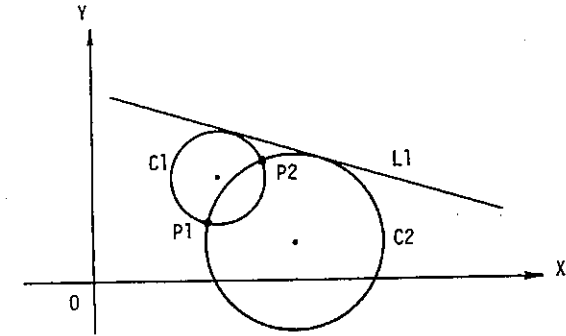
CIRCLE/TANTO,L1, $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}$, P1,RADIUS,r

CIRCLE/ $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}$, P1,TANTO, L1,RADIUS,r



TABLO 4-8 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK NOKTALARI VE DOĞRULARI KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum Doğrultu değiştirici, tanımlanan daire merkezinin diğer mümkün çözümün merkezine göre olan konumunu gösterir.	
	Örnekler	
	C1=CIRCLE/TANTO,L1,XLARGE,P1,RADIUS,3.0	
	C3=CIRCLE/YLARGE,P2,TANTO,L1,RADIUS,2.5	
4	Daire, verilen iki noktadan (P1 ve P2) geçmekte olup, verilen bir L1 doğrusuna teğettir.	CIRCLE/ $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}$, TANTO,L1,THRU,P1,P2



Yorum P1 ve P2 noktalarının her ikisi de verilen L1 doğrusu üzerinde olamaz. Genelde, mümkün olan iki çözüm vardır.

Örnekler

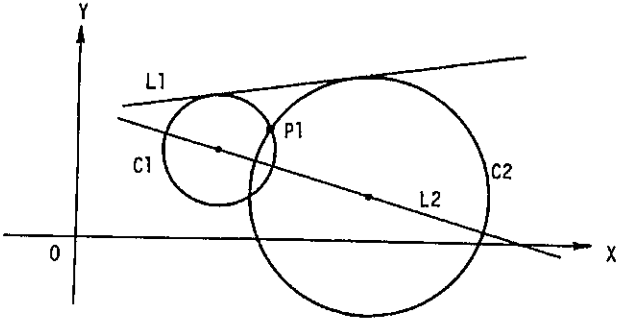
C1=CIRCLE/YLARGE,TANTO,L1,THRU,P1,P2

C2=CIRCLE/XLARGE,TANTO,L1,THRU,P1,P2

- 5 Daire verilen bir P1 noktasından geçmekte olup, verilen bir L1 doğrusuna teğettir ve merkezi de verilen diğer bir L2 doğrusu üzerindedir.

CIRCLE/ $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}$, TANTO,L1,THRU,P1,ON,L2

TABLO 4-8 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK NOKTALARI VE DOĞRULARI KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
		

Yorum Genel olarak, verilen şart için mümkün olan iki çözüm vardır.

Örnekler

C1=CIRCLE/YLARGE,TANTO,L1,THRU,P1,ON,L2

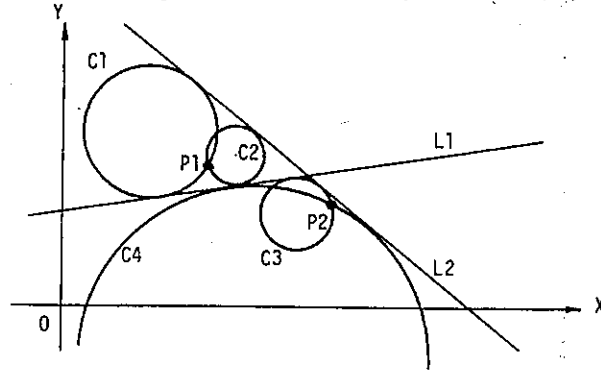
C2=CIRCLE/XLARGE,TANTO,L1,THRU,P1,ON,L2

- 6 Daire verilen bir P1 noktasından geçmekte olup, verilen iki doğruya (L1 ve L2) teğettir.

$$\text{CIRCLE/} \left\{ \begin{array}{l} \text{LARGE} \\ \text{SMALL} \end{array} \right\} \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}, \text{TANTO, \$}$$

$$\text{L1,} \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}, \text{TANTO, L2, THRU, P1}$$

Yorum Verilen iki doğru, X-Y düzlemini dört kısma böler. İki doğrultu değiştirici, X-Y düzleminin tanımlanan daireyi içine alan kısmını gösterir. Mümkün olan iki çözümden istenileni belirlemek için boyut değiştiriciler seçilebilir. Eğer P1 noktası verilen iki doğru (L1 ve L2) biri üzerinde ise, yalnızca bir çözüm mümkün olacaktır. Böyle bir durumda, LARGE veya SMALL boyut değiştiricilerinin her ikisi de aynı sonucu ortaya çıkarır.



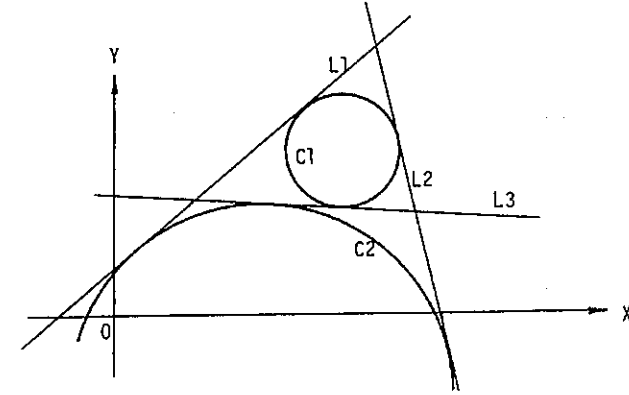
Örnekler

C1=CIRCLE/LARGE,YLARGE,TANTO,L1,YSMALL,TANTO,L2,THRU,P1

C3=CIRCLE/SMALL,YSMALL,TANTO,L1,XSMALL,TANTO,L2,THRU,P2

TABLO 4-9 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK DOĞRULARI KULLANAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Daire, verilen üç doğruya (L1,L2 ve L3) teğettir.	$\text{CIRCLE/} \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}, \text{L1,} \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}, \$$ $\text{L2,} \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}, \text{L3}$



Örnekler

C1=CIRCLE/YSMALL,L1,XSMALL,L2,YLARGE,L3

C2=CIRCLE/YSMALL,L1,XSMALL,L2,YSMALL,L3

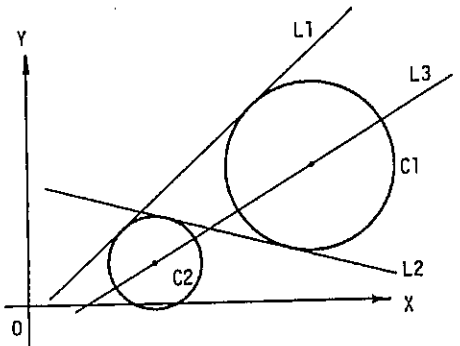
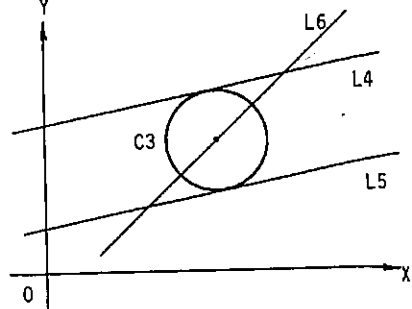
- 2 Daire verilen iki doğruya (L1 ve L2) teğet olup, merkezi verilen üçüncü bir L3 doğrusu üzerindedir.

$$\text{CIRCLE/} \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{array} \right\}, \text{TANTO, L1, TANTO, \$}$$

$$\text{L2, ON, L3}$$

Yorum Genelde, verilen şart için mümkün olan iki çözüm vardır (Şekil [a]). Bu iki çözümün merkez koordinatları karşılaştırılarak bir boyut değiştirici seçilebilir. Eğer üçüncü doğru (L3), L1 ve L2 doğruları tarafından oluşturulan açının açı ortayı ise veya verilen üç doğru birbirine paralel ve L3 doğrusu diğer ikisi arasındaki uzaklığın tam ortasında ise, sonsuz sayıda çözüm mümkün olacaktır. Bu durumda, daireyi tanımlamak için diğer deyim formatlarının kullanılması gerekir. Eğer iki teğet doğru birbirine paralel, fakat üçüncü doğru bu ikisine paralel değilse (Şekil [b]); sadece bir çözüm mevcuttur. Böyle bir durumda, yukarıda sırasıyla belirtilmiş olan değiştiricilerin tümü de aynı sonucu verir.

TABLO 4-9 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK DOĞRULARI KULLANAN DEYİMLER (devam)

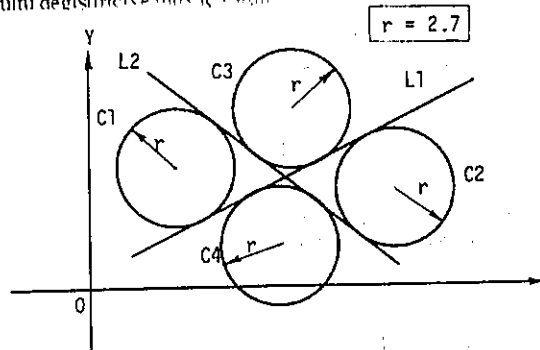
No.	Bilinen Durum	Deyim Formatı
Örnekler		
(a)		
(b)		

C1=CIRCLE/YLARGE,TANTO,L1,TANTO,L2,ON,L3
 C1=CIRCLE/XLARGE,TANTO,L1,TANTO,L2,ON,L3
 C2=CIRCLE/YSMALL,TANTO,L1,TANTO,L2,ON,L3
 C2=CIRCLE/XSMALL,TANTO,L1,TANTO,L2,ON,L3
 C3=CIRCLE/XLARGE,TANTO,L4,TANTO,L5,ON,L6
 C3=CIRCLE/XSMALL,TANTO,L4,TANTO,L5,ON,L6

- 3 Daire, kesişen iki doğruya (L1 ve L2) teğettir ve verilen bir r yarıçapına sahiptir.

CIRCLE/ { XLARGE
XSMALL
YLARGE
YSMALL } , L1, { XLARGE
XSMALL
YLARGE
YSMALL } , \$
L2, RADIUS, r

Yorum Genelde, verilen şart için mümkün olabilecek dört çözüm mevcuttur. Dolayısıyla, bu deyimde iki doğrultu değiştiriciye ihtiyaç vardır

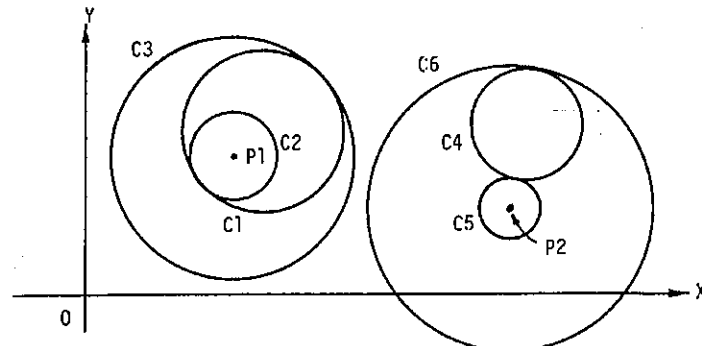


Örnekler

C1=CIRCLE/YLARGE,L1,YSMALL,L2,RADIUS,2.7
 C4=CIRCLE/XLARGE,L1,XSMALL,L2,RADIUS,2.7

TABLO 4-10 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİRE VEYA DAİRELERİ KULLANAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Daire merkezi P1 noktasında olup, verilen bir C1 dairesine teğettir.	CIRCLE/CENTER,P1, { LARGE SMALL } , TANTO,C1



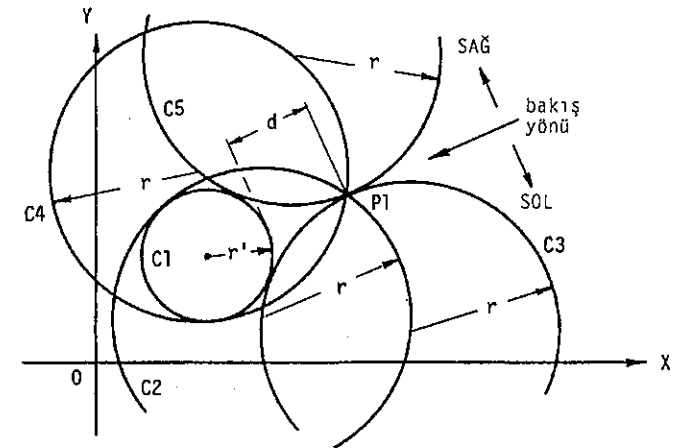
Yorum Genelde, mümkün olan iki çözüm vardır. İstenen çözümü göstermek için bir boyut değiştirici belirtilmelidir.

Örnekler

C2=CIRCLE/CENTER,P1,SMALL,TANTO,C1
 C6=CIRCLE/CENTER,P2,LARGE,TANTO,C4

- 2 Verilen bir r yarıçaplı daire, verilen bir P1 noktasından geçmekte olup, C1 dairesine teğettir.

CIRCLE/ { XLARGE
XSMALL
YLARGE
YSMALL } , { RIGHT
LEFT } , \$
TANTO, C1, THRU, P1, RADIUS, r



TABLO 4-10 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİRE VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum P1 noktası verilen C1 dairesinin dışında olduğu takdirde, genelde mümkün olabilecek dört çözüm vardır. RIGHT (SAĞ) ve LEFT (SOL) değiştiriciler P1'den C1 dairesinin merkezine doğru bakıldığında, tanımlanan daire merkezinin sırasıyla sağda ya da solda bulunabileceğini ifade eder. RIGHT ve LEFT değiştiriciler tarafından seçilmiş olan iki muhtemel çözümden istenilen çözümü göstermek için de XLARGE, XSMALL, YLARGE ve YSMALL boyut değiştiriciler kullanılır. Çeşitli şartlar altında elde edilebilecek çözüm sayısı aşağıdaki gibidir:	
	r	Çözüm sayısı
	$< d/2$	None
	$= d/2$	1
	$d/2 < r < (r' + d/2)$	2
	$= r' + (d/2)$	3
	$> r' + (d/2)$	4

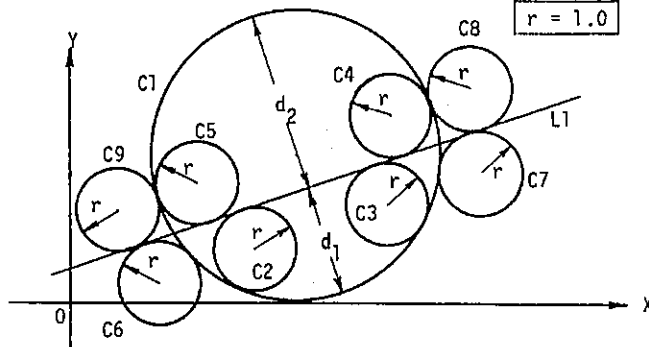
Eğer r değeri $d/2$ 'den küçükse, SD işlemcisi bir hata mesajı verecektir. Bu konuda daha ayrıntılı açıklama için, Bölüm 4.11'e bakınız.

Örnekler

C2=CIRCLE/XSMALL,LEFT,TANTO,C1,THRU,P1,RADIUS, r
 C3=CIRCLE/XLARGE,RIGHT,TANTO,C1,THRU,P1,RADIUS, r

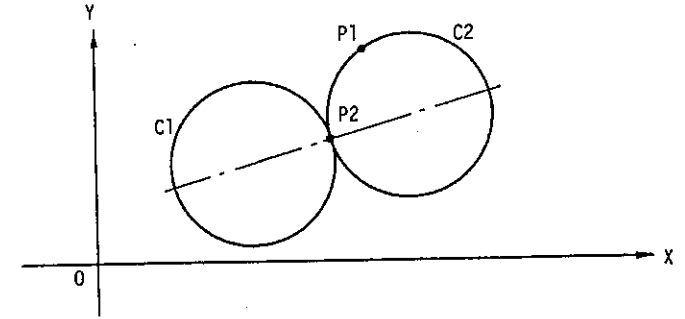
- 3 r yarıçaplı daire, verilen bir C1 dairesine ve L1 doğrusuna teğettir.

CIRCLE/ { XLARGE
XSMALL
YLARGE
YSMALL } , L1, { XLARGE
XSMALL
YLARGE
YSMALL } , \$
 { IN
OUT } , C1, RADIUS, r



TABLO 4-10 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİRE VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum Bölüm 4.11'deki açıklanaya bakınız.	
	Örnekler C2=CIRCLE/YSMALL,L1,XSMALL,IN,C1,RADIUS,1.0 C8=CIRCLE/YLARGE,L1,XLARGE,OUT,C1,RADIUS,1.0 C9=CIRCLE/YLARGE,L1,XSMALL,OUT,C1,RADIUS,1.0	
4	Daire verilen bir P1 noktasından geçmekte olup, verilen bir P2 noktasında C1 dairesine teğettir.	CIRCLE/P1,TANTO,C1,P2

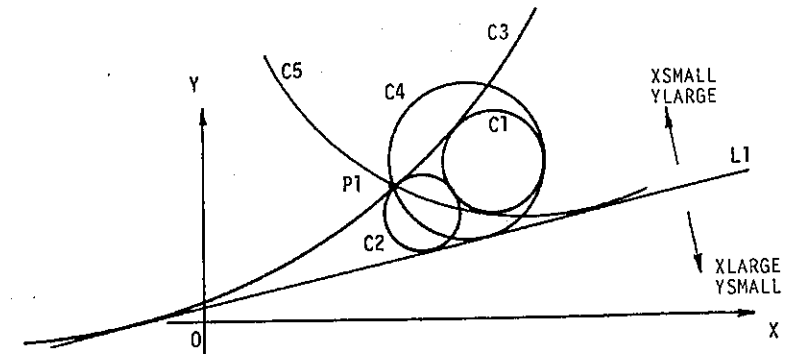


Örnek

C2=CIRCLE/P1,TANTO,C1,P2

- 5 Daire verilen bir P1 noktasından geçmekte olup, verilen bir L1 doğrusuna ve C1 dairesine teğettir. P1 ve C1, L1 doğrusunun aynı tarafında yer almaktadır.

CIRCLE / { LARGE
SMALL } , { XLARGE
XSMALL
YLARGE
YSMALL } , TANTO, \$
 L1, { IN
OUT } , TANTO, C1, THRU, P1



TABLO 4-10 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİRE VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

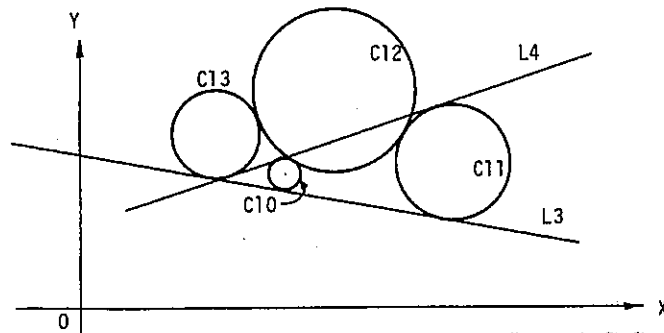
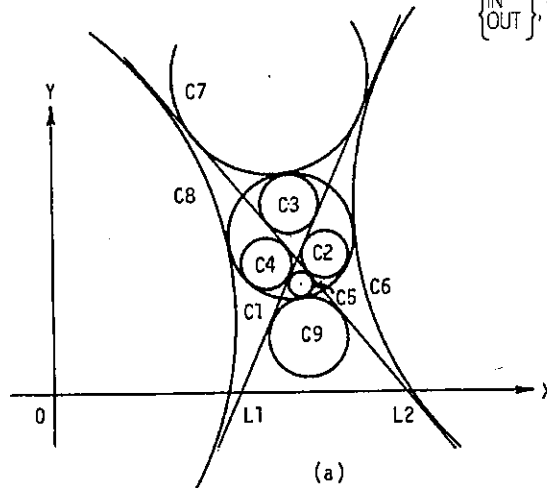
No.	Bilinen Durum	Deyim Formatı
	Yorum Eğer P1 noktası C1 veya L1 üzerinde değilse ve L1'in C1 ile hiçbir ortak noktası yoksa, genelde mümkün olabilecek dört çözüm vardır. İstenen dairenin verilen C1 dairesi ile ortak bir alanı paylaşmakta olup olmadığını göstermek için IN (iç) veya OUT (dış) değiştirici kullanılır. LARGE veya SMALL boyut değiştirici ise, istenen çözümün sırasıyla daha büyük ya da daha küçük yarıçaplı daire olduğu anlamına gelir. XLARGE, ..., ve YSMALL doğrultu değiştiricilerinin işlevi iyi tanımlanmamış olup; eğer deyimde yer alan diğer iki değiştirici üzerinde hiçbir etkisi yoktur. Bu deyimin kullanılışı ile ilgili daha ayrıntılı açıklama için, Bölüm 4.11'e bakınız.	
	Örnekler	
	C2=CIRCLE/SMALL,YLARGE,TANTO,L1,OUT,TANTO,C1,THRU,P1	
	C3=CIRCLE/LARGE,YLARGE,TANTO,L1,OUT,TANTO,C1,THRU,P1	
	C4=CIRCLE/SMALL,XSMALL,TANTO,L1,IN,TANTO,C1,THRU,P1	
	C5=CIRCLE/LARGE,XSMALL,TANTO,L1,IN,TANTO,C1,THRU,P1	

- 6 Daire, verilen iki doğruya (L1 ve L2) ve verilen bir C1 dairesine teğettir.

$$\text{CIRCLE} / \begin{Bmatrix} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{Bmatrix}, \text{TANTO}, L1, \$$$

$$\begin{Bmatrix} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{Bmatrix}, \text{TANTO}, L2, \begin{Bmatrix} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{Bmatrix}, \$$$

$$\begin{Bmatrix} \text{IN} \\ \text{OUT} \end{Bmatrix}, \text{TANTO}, C1$$



TABLO 4-10 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİRE VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum Genelde, mümkün olan sekiz çözüm vardır. Çözüm sayısı, verilen doğruların ve dairenin bağlı konumlarıyla değişir. Bazı durumlarda (örneğin Şekil (a)'daki C2 dairesi için), bu deyimde verilen C1 dairesinden önce gelen doğrultu değiştiricinin istenen dairenin tanımı üzerinde hiçbir etkisi bulunmadığına dikkat edilmelidir. Çünkü C2 dairesi, diğer üç değiştirici tarafından açık olarak tanımlanmıştır. Bu durumda, XLARGE, ..., YSMALL değiştiricilerden herhangi biri belirtilebilir. Ancak, Şekil (b)'de C12'den önce gelen doğrultu değiştirici, mümkün olan iki çözümden doğru olanı belirleyen bir seçici görevi yapar.	
	Örnekler	
	C2=CIRCLE/XLARGE,TANTO,L1,YLARGE,TANTO,L2,XLARGE,IN,TANTO,C1	
	C10=CIRCLE/YLARGE,TANTO,L3,YSMALL,TANTO,L4,XSMALL,OUT,TANTO,C12	
	C11=CIRCLE/YLARGE,TANTO,L3,YSMALL,TANTO,L4,XLARGE,OUT,TANTO,C12	

- 7 Daire verilen bir r yarıçapına sahip olup, verilen iki daireye (C1 ve C2) teğettir.

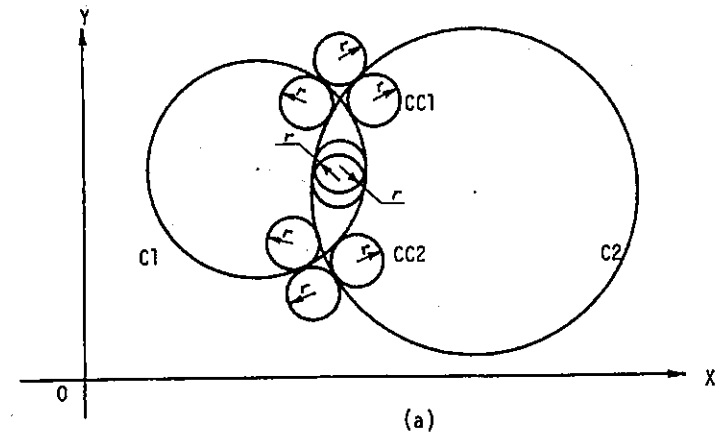
$$\text{CIRCLE} / \begin{Bmatrix} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \end{Bmatrix}, \begin{Bmatrix} \text{IN} \\ \text{OUT} \end{Bmatrix}, C1, \$$$

$$\begin{Bmatrix} \text{IN} \\ \text{OUT} \end{Bmatrix}, C2, \text{RADIUS}, r$$

Yorum Verilen şarta bağlı olarak, mümkün olabilecek çözüm sayısı 0'dan 8'e kadar değişir.

Örnekler

CC1=CIRCLE/YLARGE,OUT,C1,IN,C2,RADIUS,r
 CC2=CIRCLE/YSMALL,OUT,C1,IN,C2,RADIUS,r
 CC5=CIRCLE/YSMALL,OUT,C3,OUT,C4,RADIUS,r
 CC6=CIRCLE/YLARGE,OUT,C3,OUT,C4,RADIUS,r



TABLO 4-10 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİRE VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
8		

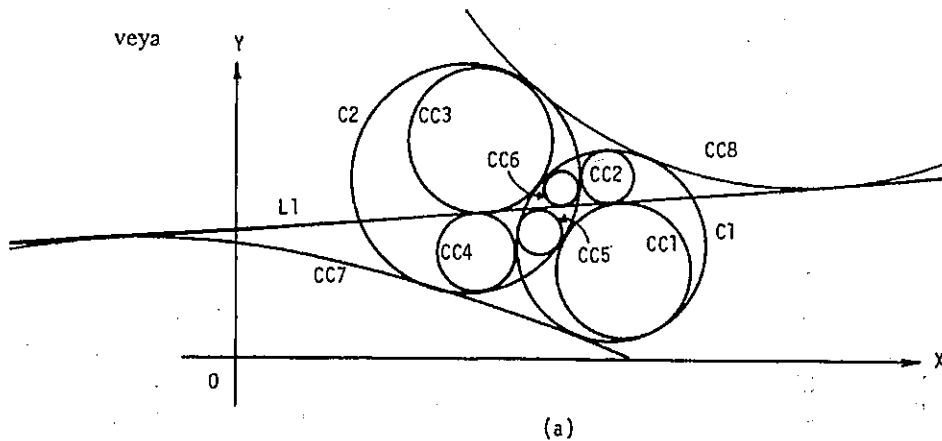
- 8 Daire, verilen iki daireye (C1 ve C2) ve bir L1 doğrusuna teğettir.

CIRCLE/ {LARGE, SMALL}, {XLARGE, XSMALL, YLARGE, YSMALL}, TANTO, \$
 L1, {IN, OUT}, TANTO, C1, {IN, OUT}, \$
 TANTO, C2

Yorum Genel bir durumda, sekiz kadar (Şekil [a]'da CC1'den CC8'ye kadar) çözüm vardır. CC5 dairesi, şu iki deyimden biri ile tanımlanabilir.

CC5=CIRCLE/LARGE,YSMALL,TANTO,L1,IN,TANTO,C1,IN,TANTO,C2

veya



TABLO 4-10 BİR DAİREYİ TANIMLAMAK İÇİN REFERANS OLARAK BİR DAİRE VEYA DAİRELERİ KULLANAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı

CC5=CIRCLE/SMALL,YSMALL,TANTO,L1,IN,TANTO,C1,IN,TANTO,C2

Dolayısıyla, bu durumda boyut değiştiricisinin hiçbir etkisi yoktur. Aşağıdaki örnekler, ihtiyaç olduğunda, bir boyut değiştiricisinin seçici rolünü göstermektedir (Şekil [b]):

C11=CIRCLE/SMALL,YSMALL,TANTO,L2,IN,TANTO,C4,OUT,TANTO,C3

C10=CIRCLE/LARGE,YSMALL,TANTO,L2,IN,TANTO,C4,OUT,TANTO,C3

Eğer C10 ve C11 aynı çapa sahipse, (Şekil [c]), bu deyim hangisinin istenen çözüm olduğunu belirleyemez. Bunun sonucu olarak da bir hata mesajı ortaya çıkar.

4.4 BİR VEKTÖRÜ TANIMLAYAN DEYİMLER

Vektör, hem belli bir büyüklüğe ve hem de belli bir yöne sahip bir doğru parçası olup; APT dilinde:

VECTOR/...

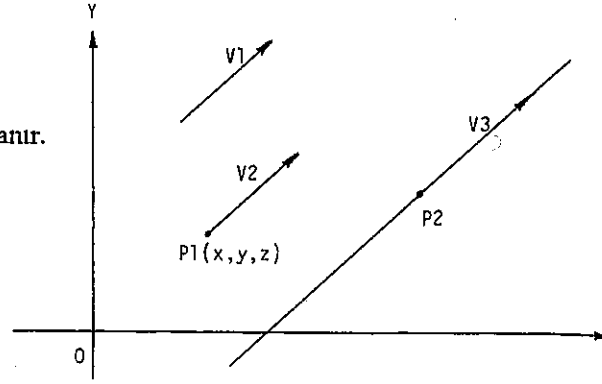
deyimi ile tanımlanır.

Matematiksel olarak, aynı büyüklüğe ve yöne sahip herhangi iki vektör aynı olarak dikkate alınır. Diğer bir deyişle, bir vektör uzayda belirli bir noktaya yerleşik değildir. NC işlemcisinde, vektör, verilen bir noktadan itibaren bir yön veya artışlı yerdeğişimi ya da her ikisini belirtmek için kullanılabilir. APT dilinde bir vektörü tanımlamak için 11 deyim formatı olup; bu formatlar Tablo 4-11'de sıra ile gösterilmektedir.

Bazı durumlarda, uzayda sabit bir başlangıç konumuna sahip bir vektörü tanımlamak gereklidir. Bu tür bir vektör, APT dilinde "nokta vektörü" adıyla bilinir (Şekil 4-7) ve

$$\text{PNTVCT} / \left\{ \begin{matrix} P1 \\ x,y,z \end{matrix} \right\}, \left\{ \begin{matrix} V1 \\ i,j,k \end{matrix} \right\}$$

deyimi vasıtasıyla tanımlanır.



Şekil 4-7 Uzayda sabit bir noktaya sahip olan vektörlerin tanımlanması (V2 sonlu, V3 ise sonsuz uzunluktadır)
 $V2 = \text{PNTVCT} / P1, V1$
 $V3 = \text{SPALIN} / P2, V1$

Burada, $P1(x,y,z)$ tanımlanacak vektörün başlangıç noktası; $V1$ ise, sırasıyla X , Y ve Z yönlerindeki i , j ve k bileşenlerine sahip olan verilen bir vektörü belirtmektedir.

Verilen bir noktadan geçen ve sonsuz uzunluğa sahip olan bir vektör, APT dilinde "uzay doğrusu" adı ile bilinir (Şekil 4-7) ve

$$\text{SPALIN} / P1, \left\{ \begin{matrix} P2 \\ V1 \end{matrix} \right\}$$

deyimiyle tanımlanır. Burada $V1$, istenen vektörün (uzay vektörünün) tanımlanmasını sağlayan bir vektördür. Eğer bu deyimde iki nokta ($P1$ ve $P2$) belirtilmişse, tanımlanan vektörün yönü $P1$ noktasından $P2$ noktasına doğrudur.

TABLO 4-11 BİR VEKTÖRÜ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Vektörün X , Y ve Z eksenindeki bileşenleri; sırasıyla x , y ve z 'dir.	VECTOR / x, y, z
2	Başlangıç ve bitiş noktaları, sırasıyla $P1(x1,y1,z1)$ ve $P2(x2,y2,z2)$ 'dir.	VECTOR / $\left\{ \begin{matrix} x1,y1,z1 \\ P1,P2 \end{matrix} \right\}$
3	Vektör X - Y düzleminde olup, verilen bir $L1$ doğrusu ile bir a açısı yapmaktadır.	VECTOR / ATANGL, $a, L1$, $\left\{ \begin{matrix} POSX \\ NEGX \\ XLARGE \\ XSMALL \\ POSY \\ NEGY \\ YLARGE \\ YSMALL \end{matrix} \right\}$
	Yorum Tanımlanan vektörün verilen doğru ile yaptığı açı a ve $(a+180^\circ)$ değerlerinden biri olup, bunlardan herhangi birisi seçilebilir. Tanımlanan vektöre ait olarak istenen yönü göstermek için de bir doğrultu değiştirici belirtilir.	

Örnek $V1$ vektörünün bileşenleri, sırasıyla:

$$x = 1 - 5 = -4$$

$$y = 5 - 1 = 4$$

$$\text{veya } z = 5 - 1 = 4$$

olduğuna göre, $V1$ 'i tanımlayan deyim:

$$V1 = \text{VECTOR} / -4,4,4$$

biçimindedir.

Örnek (No.1 deyim formatındaki şekle bakınız)
 $V1 = \text{VECTOR} / P1, P2$
 $V1 = \text{VECTOR} / 5,1,1,1,5,5$

Örnek (No.1 deyim formatındaki şekle bakınız)

$$V1 = \text{VECTOR} / P1, P2$$

$$V1 = \text{VECTOR} / 5,1,1,1,5,5$$

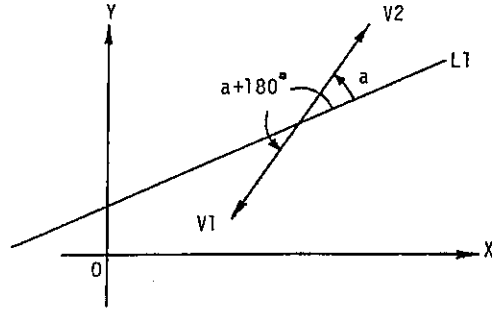
3 Vektör X - Y düzleminde olup, verilen bir $L1$ doğrusu ile bir a açısı yapmaktadır.

$$\text{VECTOR} / \text{ATANGL}, a, L1,$$

$$\left\{ \begin{matrix} POSX \\ NEGX \\ XLARGE \\ XSMALL \\ POSY \\ NEGY \\ YLARGE \\ YSMALL \end{matrix} \right\}$$

TABLO 4-11 BİR VEKTÖRÜ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
-----	---------------	---------------



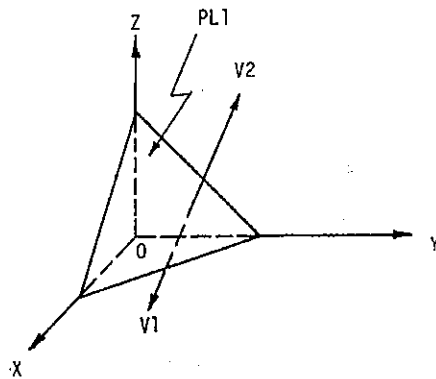
Örnekler Şekilde $a=30$ derece varsayımına göre

V1=VECTOR/ATANGL,30,L1,NEGX
 V1=VECTOR/ATANGL,30,L1,YSMALL
 V1=VECTOR/ATANGL,210,L1,NEGX
 V2=VECTOR/ATANGL,30,L1,POSX
 V2=VECTOR/ATANGL,30,L1,POSY
 V2=VECTOR/ATANGL,210,L1,POSX

- 4 Vektör verilen bir PL1 düzlemine diktir.

VECTOR/PERPTO, PL1,

POSX
 POSY
 POSZ
 NEGX
 NEGY
 NEGZ



Örnekler

V1=VECTOR/PERPTO,PL1,NEGX
 V2=VECTOR/PERPTO,PL1,POSZ

TABLO 4-11 BİR VEKTÖRÜ TANIMLAYAN DEYİMLER (devam)

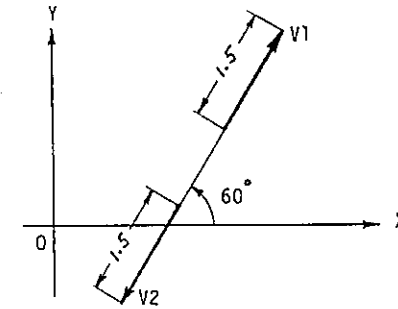
No.	Bilinen Durum	Deyim Formatı
-----	---------------	---------------

- 5 k büyüklüğündeki vektör X-Y, Y-Z veya Z-X düzlemine paralel olup; +X (X-Y düzlemi için), +Y (Y-Z düzlemi için) veya +Z (Z-X düzlemi için) eksenini ile bir a açısı yapmaktadır.

VECTOR/LENGTH, k, ATANGL, a, $\begin{cases} \text{XYPLAN} \\ \text{YZPLAN} \\ \text{ZXPLAN} \end{cases}$

Örnekler

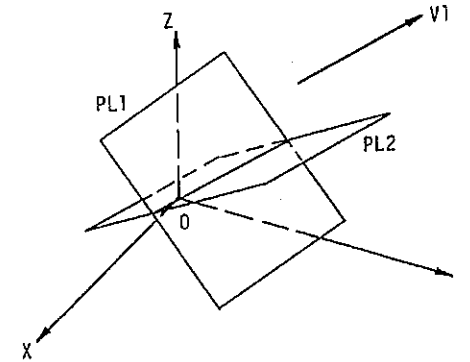
V1=VECTOR/LENGTH,1.5,ATANGL,60,XYPLAN
 V2=VECTOR/LENGTH,1.5,ATANGL,240,XYPLAN
 V2=VECTOR/LENGTH,1.5,ATANGL,-120,XYPLAN



- 6 Vektör, verilen iki düzlemin (PL1 ve PL2) kesişim doğrusuna paraleldir.

VECTOR / PARLEL, INTOF, PL1, PL2,

POSX
 POSY
 POSZ
 NEGX
 NEGY
 NEGZ

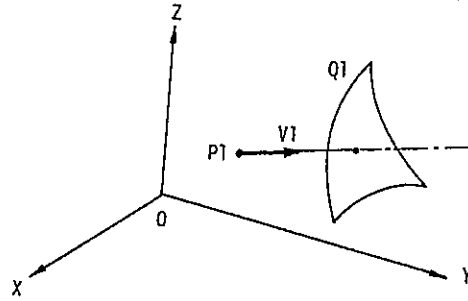


Örnek

V1=VECTOR/PARLEL,INTOF,PL1,PL2,NEGX

TABLO 4-11 BİR VEKTÖRÜ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
7	Vektör, verilen bir P1 noktasından geçen ve Q1 ikinci dereceden yüzeye * dik olan bir birim vektördür.	VECTOR / $\begin{Bmatrix} \text{POSX} \\ \text{POSZ} \\ \text{NEGX} \\ \text{NEGY} \\ \text{NEGZ} \end{Bmatrix}$, PERPTO,Q1,P1

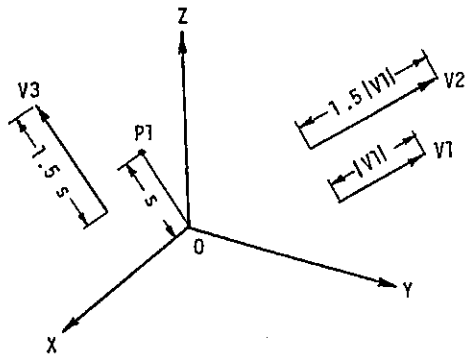
**Örnek**

V1=VECTOR/POSY,PERPTO,Q1,P1

- 8 Vektör, verilen bir V1 vektörünün n skalar ile çarpımıdır.

VECTOR/n, TIMES, $\begin{Bmatrix} P1 \\ V1 \end{Bmatrix}$

Yorum Bir vektör yerine bir P1 noktası verilmişse, referans vektör, koordinat sisteminin başlangıç noktasından P1 noktasına uzanır.

**Örnekler**

V2=VECTOR/1.5,TIMES,V1

V2=VECTOR/1.5,TIMES,P1

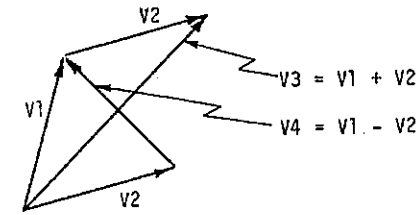
* İkinci dereceden yüzeylerin tanımı için, Bölüm 4.7'ye bakınız.

TABLO 4-11 BİR VEKTÖRÜ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
9	Vektör; yönü, verilen bir V1 vektörü veya P1(x,y,z) noktası tarafından tanımlanan bir birim vektördür.	VECTOR/UNIT, $\begin{Bmatrix} X,Y,Z \\ P1 \\ V1 \end{Bmatrix}$

- 10 Vektör, verilen iki vektörün (V1 ve V2) vektörel toplamı veya farkıdır.

VECTOR/W1, $\begin{Bmatrix} \text{PLUS} \\ \text{MINUS} \end{Bmatrix}$, V2

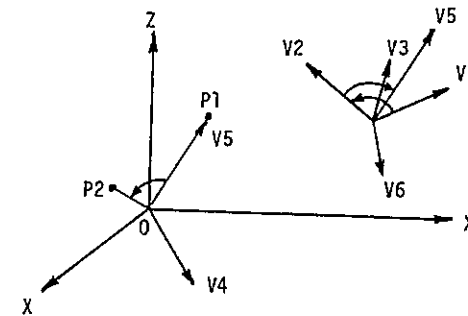
**Örnekler**

V3=VECTOR/W1,PLUS,V2

V4=VECTOR/W1,MINUS,V2

- 11 Vektör, verilen iki vektörün (V1 ve V2) vektörel çarpımıdır.

VECTOR / $\begin{Bmatrix} V1 \\ P1 \end{Bmatrix}$, CROSS, $\begin{Bmatrix} V2 \\ P2 \end{Bmatrix}$



Yorum İki vektör yerine iki nokta verilmişse; bu iki vektör, koordinat sisteminin başlangıç noktasını kendi başlangıç noktası olarak alan ve verilen noktaları da sırasıyla kendi bitiş noktası olarak kabul eden vektörlerdir.

Örnekler

V3=VECTOR/W1,CROSS,V2

V4=VECTOR/P1,CROSS,P2

V6=VECTOR/W2,CROSS,P1

4.5 DÜZLEMSEL EĞRİLERİ (ELİPS, HİPERBOL, GCONIC VE LCONIC) TANIMLAYAN DEYİMLER

Elips ve hiperbol, mühendislik tasarımında yaygın olarak kullanılan iki düzlem eğrisidir. Genelde, elips, bir düzlemle bir koninin veya bir dairesel silindirin kesişimi, ya da bir dairenin, bu daireyi ihtiva eden düzleme paralel olmayan bir düzlem üzerindeki izdüşümüdür. Elips, hareketli bir P2 noktasının, düzlemde bulunan iki sabit noktadan (P2 ve P3) uzaklıkları toplamı daima sabit kalacak biçimde hareket ederek meydana getirdiği geometrik yer alan bir düzlem eğrisi olarak da tanımlanır (Şekil 4-8'de S1'e bakınız).

X-Y düzlemindeki bir elipse ait matematiksel ifadenin standart biçimi:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

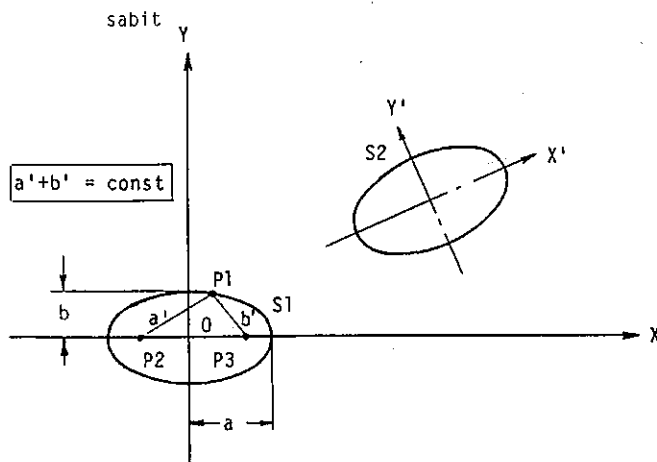
olup; burada, a ve b sırasıyla elipsin büyük ve küçük eksenlerinin yarı mesafeleridir. X-Y düzleminde yer alan ve yukarıdaki ifade ile tanımlananın dışındaki herhangi bir elips (örneğin Şekil 4-8'de S2), koordinat sisteminin dönüşümü suretiyle de tanımlanabilir.

Bir elips;

ELLIPS/...

deyimiyle tanımlanır. APT dilinde, bir elipsi tanımlayan iki deyim formatı mevcut olup; bunlar Tablo 4-12'de sıra ile gösterilmektedir.

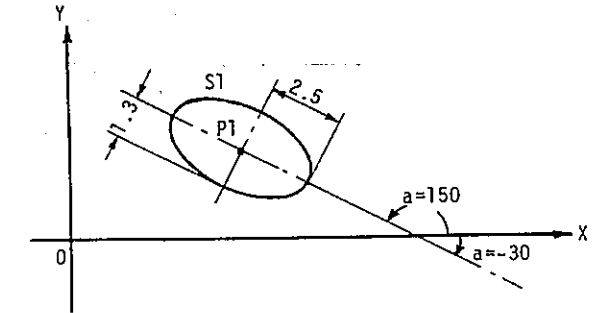
Hiperbol ise; hareketli bir P1 noktasının aynı düzlemdeki verilen iki noktadan (odak noktalarından) (P2 ve P3) uzaklıkları farklı daima sabit kalacak biçimde hareket ederek, meydana getirdiği geometrik yerdir (Şekil 4-9).



Şekil 4-8 Bir elipsin tanımı

TABLO 4-12 BİR ELİPSİ TANIMLAYAN DEYİMLER

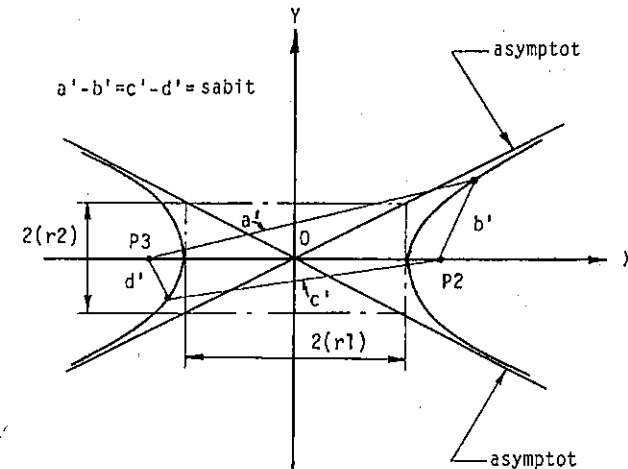
No.	Bilinen Şart	Deyim Formatı
1	Elipsin büyük ve küçük eksenlerinin yarıları sırasıyla R1 ve R2, merkezi ise verilen bir P1 noktasında olup; büyük eksen +X eksenine ile bir α açısı yapmaktadır.	ELLIPS/CENTER,P1,R1,R2, α
2	Elips, verilen bir PL1 düzlemi ile verilen bir S1 silindiri veya konisinin kesişimidir.	ELLIPS/ANTOF,PL1,S1



Örnekler

S1=ELLIPS/CENTER,P1,2.5,1.3,150
S1=ELLIPS/CENTER,P1,2.5,1.3,-30

Yorum Eğer S1 bir silindir ise, PL1 düzlemi silindir eksenine paralel olmamalı, eğer S1 bir koni ise, PL1 düzlemi ile koni eksenindeki açı koni tepe açısının yarısından büyük olmamalıdır.



Şekil 4-9 Hiperbol ve parametreleri

X-Y düzleminde bulunan ve enine eksenini X eksenine olan bir hiperbola ait matematiksel ifadenin standart biçimi:

$$\frac{x^2}{(r1)^2} - \frac{y^2}{(r2)^2} = 1$$

olup; burada r1 ve r2 sırasıyla enine ve eşlenik eksenlerdir. Koordinat sisteminin dönüşümü suretiyle, herhangi bir konumdaki ve herhangi bir yöndeki bir hiperbol yukarıda gösterilen ifade ile tanımlanabilir. Böylece, APT dilinde bir hiperbol için verilen şart, koordinat sisteminin başlangıç noktası dışında bir başka nokta olabilen P1 merkez noktası, sırasıyla r1 v r2 ile gösterilen yarı enine eksen ve yarı eşlenik eksen, ve enine eksenin X eksenine yaptığı açılarından oluşur.

Deyim formatı:

HYPERB/CENTER,P1,r1,r2,a

biçimindedir.

X-Y düzlemindeki bir eğrinin ikinci dereceden bir eşitlikle tanımlandığı durumlar vardır. APT dilinde bu tür bir eğri genel konik adı ile bilinir ve daire, elips, hiperbol ve paraboller kapsar. Bu eğri:

GCONIC/.....

deyimiyle tanımlanabilir.

Genel konikleri tanımlamak için APT'de üç deyim formatı mevcut olup, bunlar Tablo 4-13'de sırasıyla belirtilmiştir.

Bazan, dizayn çalışmasında, yukarıda açıklananların dışındaki eğriler (örneğin daha yüksek dereceden eşitlikler vasıtasıyla tanımlanan eğriler) kullanılır. Bir alternatif yöntem olarak, bir eğri üzerinde birkaç ayrı nokta ve eğrinin bu noktaların bazılarındaki eğimleri verilebilir. Diğer durumlarda, plastik eğri şablonları kullanılarak istenen eğri çizilebilir. Söz konusu eğrinin basit bir matematiksel ifade ile tarif edilememesi de mümkündür. Bununla beraber, belirli birkaç noktanın koordinatlarını ve bazı noktadaki eğimleri biliriz. Dolayısıyla, yapmamız gereken şey, verilen veya hesaplanmış olan noktalardan geçen ve belirli noktalardaki eğimlere uyan hassas bir eğri elde etmektir.

TABLO 4-13 : BİR GENEL KONİK ŞEKLİ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Eğri; $ax^2 + bxy + cy^2 + dx + ey + f = 0$ eşitliği ile tanımlanmaktadır.	GCONIC/a,b,c,d,e,f
	Örnek Eğer bir C1 eğrisi $3x^2 + 2xy + 4y^2 + 1.5x - 5 = 0$ eşitliği ile tanımlanırsa; bu eğriyi tanımlayan APT deyimini: C1=GCONIC/3,2,4,1.5,0,-5	
2	Eğri; $y = px + q \pm (rx^2 + sx + t)^{\frac{1}{2}}$ eşitliği ile tanımlanmaktadır.	GCONIC/p,q,r,s,t
3	Eğri; $x = py + q \pm (ry^2 + sy + t)^{\frac{1}{2}}$ eşitliği ile tanımlanmaktadır.	GCONIC/p,q,r,s,t,FUNOFY

Matematiksel olarak bu, verilen şartları (noktalar ve eğimler) sağlamak üzere³ uygun enterpolasyon fonksiyonlarının kullanımıyla, söz konusu eğrinin matematik formülasyonunu yaratmaktan ibarettir. APT dilinde bu tür eğriler yüksek (loft) konik olarak bilinir ve:

LCONIC/.....

deyimi ile tanımlanır.

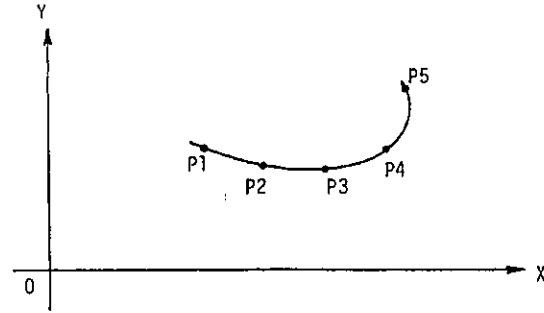
Bu konuda üç deyim formatı mevcuttur (Tablo 4-14). Bir yüksek (loft) konik bir uçta verilen bir eğri veya doğruya teğet olduğu takdirde, eğim devamlılığını sağlamak için No.2 veya No.3 formatı kullanmak daha iyidir (Şekil 4-10).

4.6 BİR DÜZLEMİ TANIMLAYAN DEYİMLER

Düzlemler, freze kesicisinin hareket güzergahını tanımlamak için en sık kullanılan geometrik öğelerdir. Kullanıcının verilen çeşitli şartlar bağlı olarak bir düzlemi tanımlayabilmesini sağlamak üzere, APT dilinde 14 deyim formatı mevcuttur (Tablo 4-14).

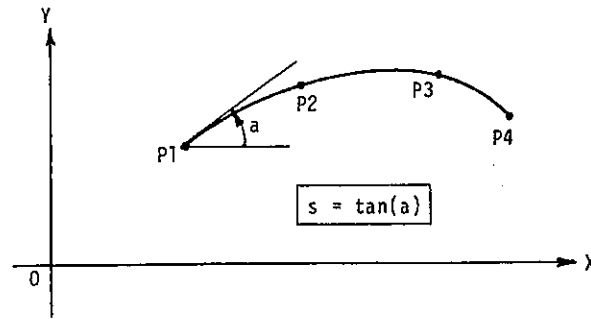
TABLO 4-14 X-Y DÜZLEMİNDE LOFT KONİKLERİ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Eğri, belirtilen sırada $P1(x1,y1)$, $P2(x2,y2)$, $P3(x3,y3)$, $P4(x4,y4)$ ve $P5(x5,y5)$ olmak üzere verilen beş noktadan geçmektedir. $P1$ ve $P5$, eğrinin uç noktalarıdır.	LCONIC/SPT $\{x1, y1, x2, y2, x3, y3, x4, y4, x5, y5\}$ $\{P1, P2, P3, P4, P5\}$



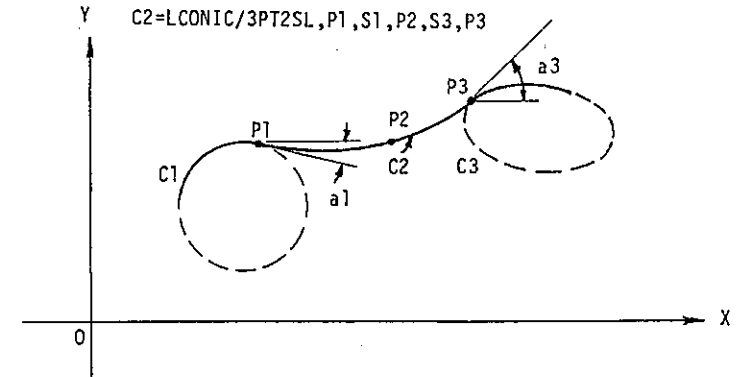
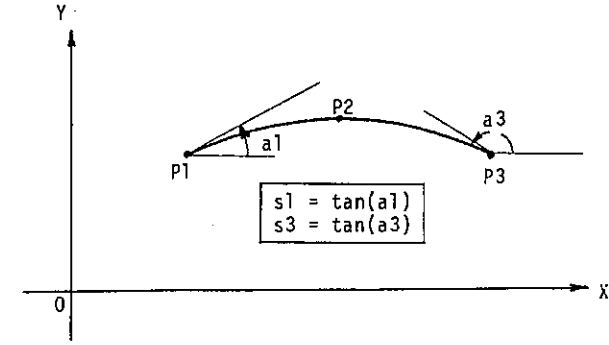
- 2 Eğri, verilen birinci noktada, ($P1$), s eğimine sahip olup; belirtilen sırada $P1(x1,y1)$, $P2(x2,y2)$, $P3(x3,y3)$ ve $P4(x4,y4)$ olmak üzere verilen dört noktadan geçmektedir. $P1$ ve $P4$ başlangıç ve bitiş noktalarıdır.

LCONIC/4PT1SL,
 $\{x1, y1, s, x2, y2, x3, y3, x4, y4\}$
 $\{P1, s, P2, P3, P4\}$



TABLO 4-14 X-Y DÜZLEMİNDE KONİKLERİ TANIMLAYAN DEYİMLER (devam)

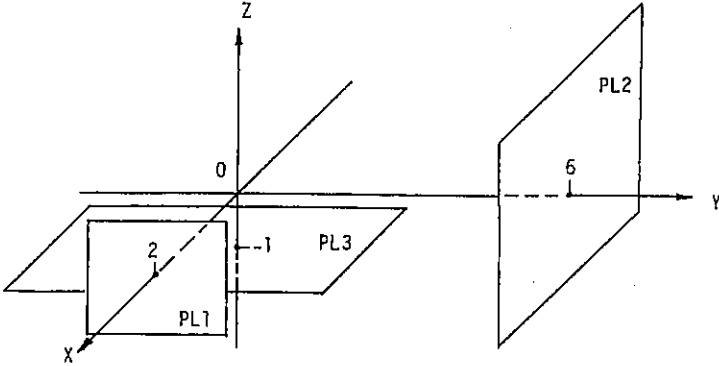
No.	Bilinen Durum	Deyim Formatı
3	Eğri, uç noktalarda ($P1$ v $P3$) sırasıyla $S1$ ve $S3$ eğimlerine sahip olup; belirtilen sırada $P1(x1,y1)$, $P2(x2,y2)$ ve $P3(x3,y3)$ olmak üzere verilen üç noktadan geçmektedir.	LCONIC/3PT2SL, $\{x1, y1, s1, x3, y3, s3, x2, y2\}$ $\{P1, s1, P3, s3, P2\}$



Şekil 4-10 Bir loft koniğinin üç nokta ve iki eğim vasıtasıyla tanımlanması, $P1$, $P2$ ve $P3$ verilen üç noktadır. $P1$ ve $P3$, aynı zamanda, $C2$ loft koniğinin verilen $C1$ ve $C3$ eğrilerine sırasıyla $S1 = \tan(a1)$ ve $S3 = \tan(a3)$ eğimleriyle teğet konumunda olduğu noktalarıdır.

TABLO 4-15 BİR DÜZLEMİ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Düzlem $ax + by + cz = d$ eşitliği ile tanımlanmaktadır	PLANE/ a,b,c,d



Örnekler

PL1=PLANE/1,0,0,2

PL2=PLANE/0,1,0,6

PL3=PLANE/0,0,1,-1

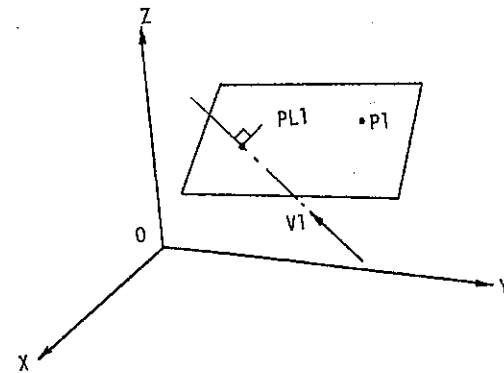
- 2 Düzlem verilen üç noktadan (P1, P2 ve P3) geçmektedir.

PLANE/P1,P2,P3

Yorum P1, P2 ve P3; önceden tanımlanmış olan üç noktanın sembolleridir.

- 3 Düzlem, verilen bir P1 noktasından geçmekte olup; bir V1 vektörüne diktir.

PLANE/P1,PERPTO,V1

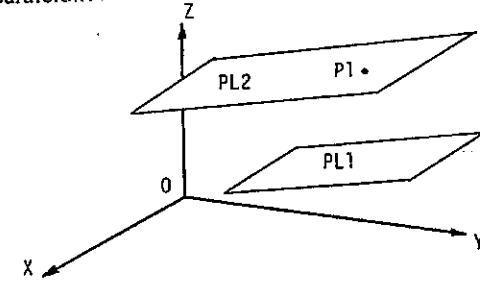


Örnek

PL1=PLANE/P1,PERPTO,V1

TABLO 4-15 BİR DÜZLEMİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
4	Düzlem, verilen bir P1 noktasından geçmekte olup; verilen bir PL1 düzlemine paraleldir.	PLANE/P1,PARLEL,PL1

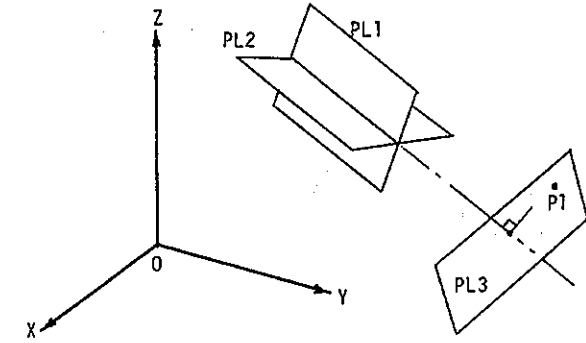


Örnek

PL2=PLANE/P1,PARLEL,PL1

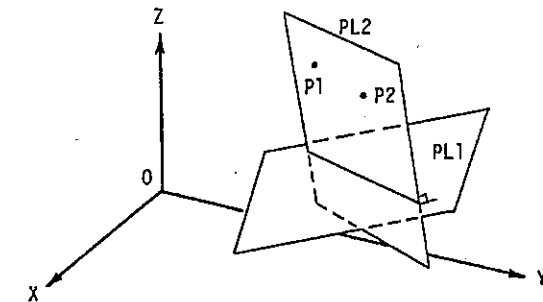
- 5 Düzlem, verilen bir P1 noktasından geçmekte olup; birbirine paralel olmayan verilen iki düzleme (PL1 ve PL2) diktir.

PLANE/P1,PERPTO,PL1,PL2



- 6 Düzlem, verilen iki noktadan (P1 ve P2) geçmekte olup; bir PL1 düzlemine diktir.

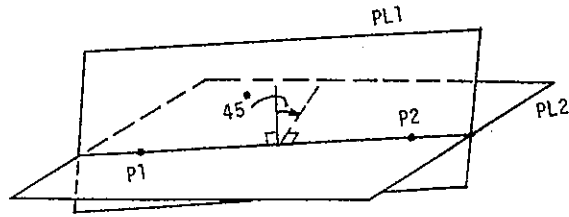
PLANE / {PERPTO,PL1,P1,P2} / {P1,P2,PERPTO,PL1}



TABLO 4-15 BİR DÜZLEMİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum Eğer P1 ve P2 noktalarından geçen doğru, verilen düzleme dik ise; sonsuz sayıda çözüm mevcut olacaktır. Böyle bir durumda, diğer deyim formatlarının kullanılması gerekir.	

- 7 Düzlem, bir PL1 düzleminde verilen iki noktadan (P1 ve P2) geçmekte ve düzlem ile bir a açısı yapmaktadır.



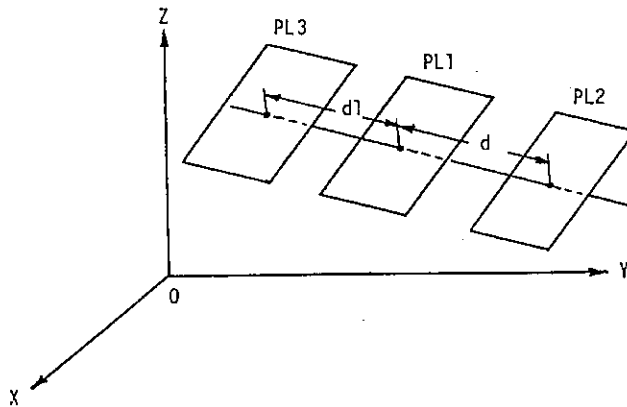
Yorum a açısı, P1 ve P2 noktalarını birleştiren doğruya dik bir düzlemde, verilen PL1 düzleminde tanımlanan PL2 düzlemine doğru ölçülür. P1 noktasından P2 noktasına doğru bakıldığında saat dönme yönünün ters yönünde ölçülen açı, pozitif değerlidir.

Örnekler

PL2=PLANE/P1,P2,ATANGL,45,PL1
PL2=PLANE/P2,P1,ATANGL,-45,PL1

- 8 Düzlem, verilen bir PL1 düzlemine paralel düzlemden d uzaklığındadır.

PLANE/PARLEL,PL1,
 $\left. \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \\ \text{ZLARGE} \\ \text{ZSMALL} \end{array} \right\}$



TABLO 4-15 BİR DÜZLEMİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum Bu deyimdeki doğrultu değiştirici, düzlemlere dik olan bir doğru boyunca verilen düzlemde tanımlanan düzleme hareket yönünü gösterir.	

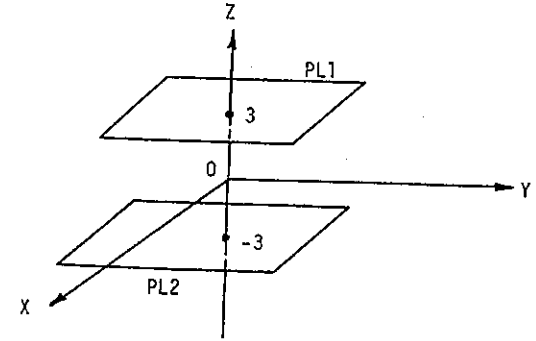
Örnek

PL2=PLANE/PARLEL,PL1,YLARGE,d
PL3=PLANE/PARLEL,PL1,YSMALL,d1

- 9 Düzlem; X-Y, Y-Z veya Z-X koordinat düzlemine paralel ve ondan bir d uzaklığındadır.

PLANE/ $\left\{ \begin{array}{l} \text{XYPLAN} \\ \text{YZPLAN} \\ \text{ZXPLAN} \end{array} \right\}$, $\{d\}$

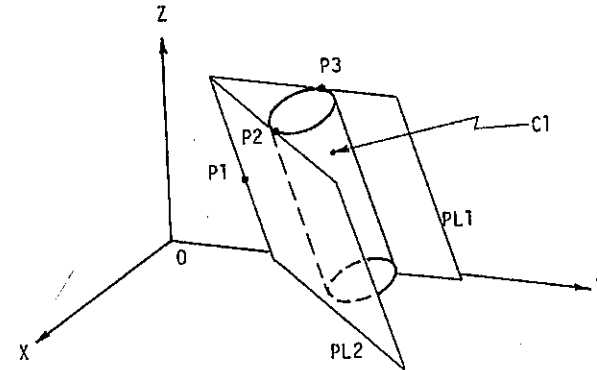
Yorum Eğer bu deyimde d yer almıyorsa, değeri sıfır olarak kabul edilir. Bu durumda, tanımlanan düzlem belirtilen koordinat düzleminin kendisidir. Bir pozitif d değeri, tanımlanan düzlemin verilen koordinat düzleminin pozitif tarafında; negatif d değeri ise negatif tarafta olduğunu ifade eder.

**Örnekler**

PL1=PLANE/XYPLAN,3
PL2=PLANE/XYPLAN,-3

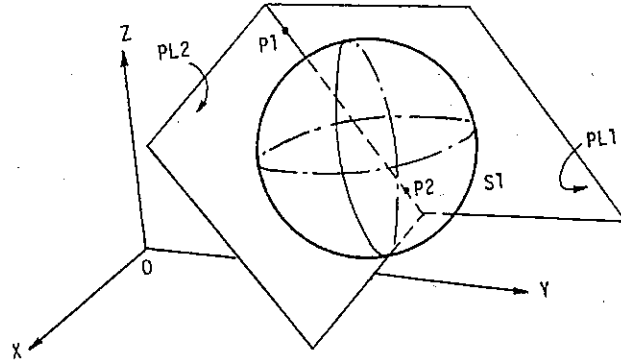
- 10 Düzlem, verilen bir P1 noktasından geçmekte olup; bir C1 silindirin teğettir.

PLANE/P1, $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \\ \text{ZLARGE} \\ \text{ZSMALL} \end{array} \right\}$,TANTO,C1



TABLO 4-15 BİR DÜZLEMİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Yorum Doğrultu değiştirici, silindir eksenine dik düzlemde bulunan iki teğet noktasını (P2 ve P3) karşılaştırmak suretiyle, daire tanım deyimine ait doğrultu değiştirici ile aynı yöntemle seçilebilir. Referans noktası P1; bakış yönü ise P1 noktasından C1 silindirine doğrudur.	
	Örnekler PL1=PLANE/P1,XSMALL,TANTO,C1 PL2=PLANE/P1,XLARGE,TANTO,C1	
11	Düzlem, verilen bir P1 noktasından geçmekte ve bir C1 silindirine dik konumdadır (yani düzlem, verilen silindirin eksenine diktir).	PLANE/P1,PERPTO,C1
12	Düzlem; verilen bir C1 küresine, küre üzerinde verilen bir P1 noktasına teğettir.	PLANE/P1,TANTO,C1
13	Düzlem, verilen iki noktadan (P1 ve P2) geçmekte olup, bir S1 küresine teğettir..	PLANE/ $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \\ \text{ZLARGE} \\ \text{ZSMALL} \end{array} \right\}$ TANTO,S1,THRU,P1,P2



Yorum Konumsal değiştirici, kürenin merkezine göre teğet değme noktasının konumunu gösterir.

TABLO 4-15 BİR DÜZLEMİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Örnekler PL1=PLANE/XSMALL,TANTO,S1,THRU,P1,P2 PL2=PLANE/YSMALL,TANTO,S1,THRU,P1,P2	

4.7 İKİNCİ DERECEDEDEN YÜZEYLERİ, SİLİNDİRLERİ VE DÖNER YÜZEYLERİ TANIMLAYAN DEYİMLER

Bir parçanın veya makina elemanının mümkün olan en basit biçimde tasarımının gerekli olmasına rağmen, bir tasarım mühendisinin istenen profilin parçaları olarak kavisli yüzeyler kullanmak zorunda olduğu birçok durum vardır. Matematiksel olarak bir yüzey, sürekli olarak

$$f(x,y,z) = 0$$

ilişkisini koruyarak uzayda hareket eden bir noktanın (x,y,z) geometrik yeridir (Şekil 4-11).

Mühendislik tasarımında yaygın olarak kullanılan yüzeyler grubu;

$$ax^2 + by^2 + cz^2 + fyz + gzx + hxy + px + qy + rz + d = 0 \quad (4.1)$$

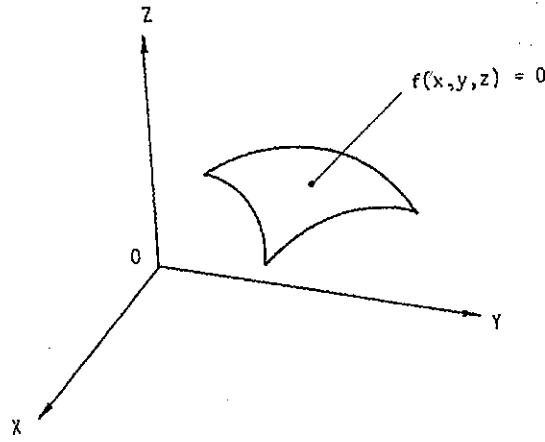
biçiminde ikinci dereceden bir genel eşitlik vasıtasıyla tanımlanan ve pramidial yüzeyler olarak bilinen yüzeylerden oluşur. Bir ikinci dereceden yüzeyi tanımlayan APT deyimini;

$$\text{QADRIC}/a,b,c,f,g,h,p,q,r,d$$

olup; burada a,b,...,d; 4.1 no.lu eşitlikte belirtilen katsayılardır.

APT dilinde tanımlanan bir ikinci dereceden yüzeyin kesinlikle bir gerçek yüzey olması (yani, 4.1 no.lu eşitliği sağlayan x,y ve z koordinatlarının kesinlikle gerçek sayılar olması) şarttır. Bir genel ikinci dereceden yüzey eşitliği, kanonik biçim olarak bilinen bir standart biçimde aşağıdaki gibi yazılabilir:

$$ax^2 + by^2 + cz^2 + 2f'yz + 2g'zx + 2h'xy + 2p'x + 2q'y + 2r'z + d = 0 \quad (4.2)$$



Şekil 4-11. Bir yüzeyin tanımı

Burada,
 $f' = f/2$;
 $g' = g/2$;
 $h' = h/2$;
 $p' = p/2$;
 $q' = q/2$;
 $r' = r/2$; dir.

$$I_1 = a + b + c$$

$$I_2 = \begin{vmatrix} a & h' & g' \\ h' & b & f' \\ g' & f' & c \end{vmatrix} + \begin{vmatrix} b & f' & c \\ f' & c & g' \\ g' & c & a \end{vmatrix} + \begin{vmatrix} c & g' & a \\ g' & a & h' \\ h' & a & b \end{vmatrix}$$

$$I_3 = \begin{vmatrix} a & h' & g' & p' \\ h' & b & f' & q' \\ g' & f' & c & r' \\ p' & q' & r' & d \end{vmatrix}$$

$$I_4 = \begin{vmatrix} a & h' & g' & p' \\ h' & b & f' & q' \\ g' & f' & c & r' \\ p' & q' & r' & d \end{vmatrix}$$

biçiminde ifade edilen dört nicelik, koordinat sisteminin transformasyonlarına göre değişmez ve I_4 değerine 4.2⁵ no.lu eşitliğin diskriminant'ı denir 4.2 no.lu ikinci dereceden eşitlikle tarif edilen yüzeylerin tipi, Tablo 4.16'da gösterildiği gibi, I_1 , I_2 , I_3 ve I_4 değerleri vasıtasıyla belirlenebilir.

Genelde, sanal yüzeyin haricindeki bir yüzeyi tanımlayan ikinci dereceden bir eşitlik koordinat sisteminin transformasyonu vasıtasıyla basitleştirilerek; genel

olarak bir elipsoid veya hiperboloid'i tanımlayan:

$$a'x^2 + b'y^2 + c'z^2 + d' = 0 \quad (4.3)$$

biçimine; veya genellikle bir eliptik ya da hiperbolik paraboloid'i tanımlayan aşağıdaki standart biçimlerden birine dönüştürülebilir:

$$\begin{aligned} a''x^2 + b''y^2 + r''z &= 0 \\ b''y^2 + c''z^2 + s''x &= 0 \\ c''z^2 + a''x^2 + r''y &= 0 \end{aligned}$$

NC işlemcisi tarafından işlenebilen ikinci dereceden yüzeyler sıra ile Şekil 4-12(a) ve (b)'de gösterilmiştir. Şekillerden görülebileceği gibi, bunlar, 4.3 ve 4.4 no.lu eşitlikler tarafından tanımlananlar ile aynı yüzeylerdir.

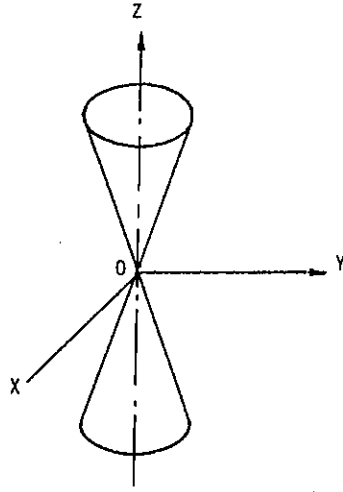
Nokta, doğru ve düzlem gibi bozulmuş ikinci dereceden yüzeylerin bazılarını tanımlamak için bir QADRIC deyim kullanılmamalı; bunların yerine, önceki kısımlarda açıklanmış olan nokta, doğru ve düzlemleri tanımlayan deyimler seçilmelidir.

Mühendislik tasarımında özellikle ilginç iki yüzey tipi mevcut olup; bunlar silindirik yüzeyler ve döner yüzeylerdir.

TABLO 4-16 İKİNCİ DERECEDEDEN YÜZEYLERİN SINIFLANDIRILMASI

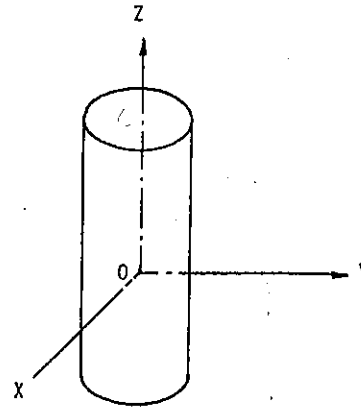
I_4	$I_3 \neq 0$		$I_3 = 0$		Sanal koni Koni
	$I_1 \cdot I_3 > 0, I_2 > 0$	$I_1 \cdot I_3 \leq 0, I_2 \leq 0$	I_4	I_2	
<0	Elipsoid	İki kanatlı (yapraklı) hiperboloid	<0	>0	Eliptik paraboloid
>0	İzafi elipsoid	Bir yapraklı hiperboloid	>0	<0	Hiperbolik paraboloid
-0	İzafi koni	Koni	-0	Herhangi bir	Silindir* veya düzlem çiftleri

* Bu listede belirtilen silindir, bir eliptik, parabolik veya dairesel silindir olabilen bir silindirik yüzeydir.



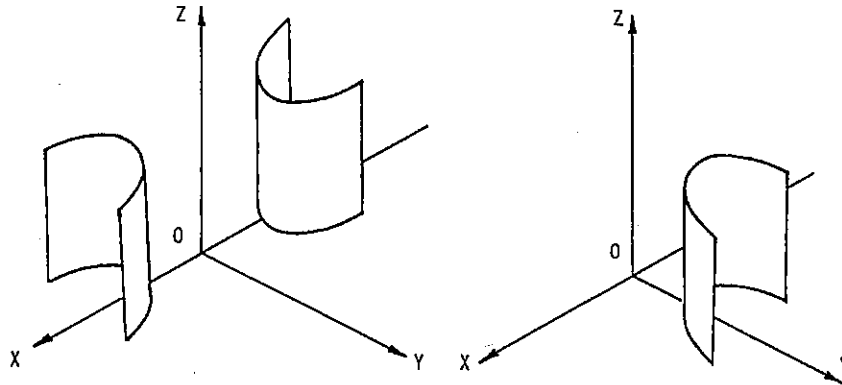
Eliptik koni:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0$$



Eliptik silindir:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$



Hiperbolik silindir:

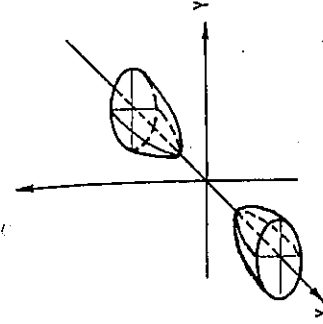
$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

Parabolik silindir:

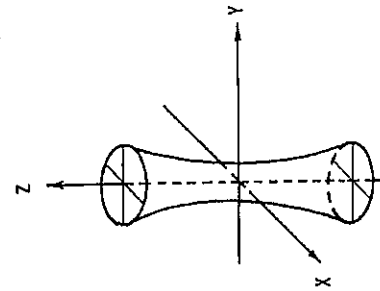
$$x^2 = py$$

(a)

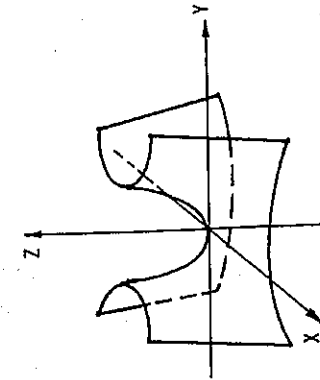
Şekil 4-12 (a) APT'de izin verilen bozalmış ikinci dereceden yüzeyler, (b) APT'de tanımlanan uygun ikinci dereceden yüzeyler.



İki yapraklı hiperboloid:
 $(x/a)^2 - (y/b)^2 - (z/c)^2 = 1$

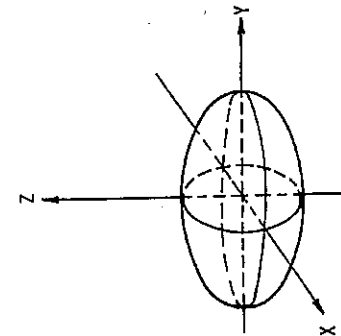


Tek yapraklı hiperboloid:
 $(x/a)^2 + (y/b)^2 - (z/c)^2 = 1$

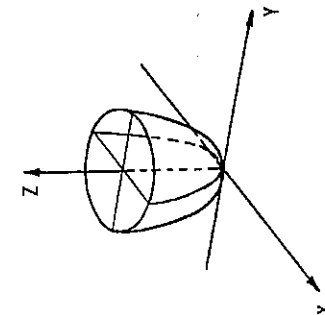


Hiperbolik paraboloid:
 $(x/a)^2 - (y/b)^2 = 2cz$

(b)



Elipsoid: $(x/a)^2 + (y/b)^2 + (z/c)^2 = 1$



Eliptik paraboloid:
 $(x/a)^2 + (y/b)^2 = 2cz$

Bir silindirik yüzey; verilen bir düzlemde bulunan ve doğrultman verilen bir eğri boyunca, bu eğriyi ihtiva eden düzlemde olmayan bir doğrunun yer değiştirilmesiyle meydana getirilir. Eğer koordinat sistemi; X, Y veya Z eksenlerinden biri oluşturulan doğru paralel olacak şekilde döndürülürse; silindirik yüzey denklemi, sadece iki değişken ihtiva edecek biçimde aşağıdaki gibi basitleştirilebilir:

$$f(x,y) = 0$$

$$f(x,z) = 0$$

veya

$$f(y,z) = 0$$

QADRIC deyim vasıtasıyla tanımlanan silindirlerin yanısıra; APT dilinde, bir dairesel silindiri tanımlamak için:

CYLNDR/.....

deyimi (Tablo 4-17); bir çizelgelenmiş silindiri (yani; doğrultmanı, verilen bir noktalar ve eğimler kümesi ile tanımlanan bir silindir [Şekil 4-13]) tanımlamak için de:

TABCYL/.....

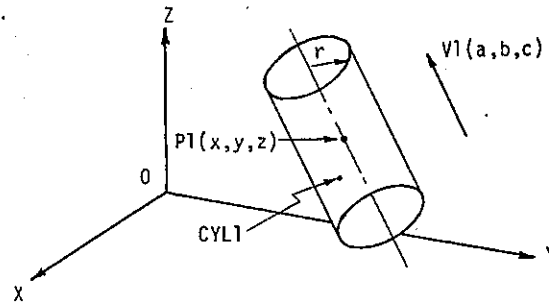
deyimi mevcuttur. TABCYL deyimi, evvelce belirtilmiş olan deyimlerle tanımlanamayan bir silindirik yüzeyi tanımlayabilmemizi mümkün kılar. TABCYL deyiminin formatlarına ait açıklama ve kullanımı, Tablo 4-18 ve 4-19'da verilmiştir.

TABLO 4-17 DAİRESEL SİLİNDİRLERİ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
-----	---------------	---------------

- 1 Silindir r yarıçaplı olup; eksenini, verilen bir $P1(x,y,z)$ noktasından geçmekte ve bir $V1(a,b,c)$ birim vektörüne paralel konumda bulunmaktadır.

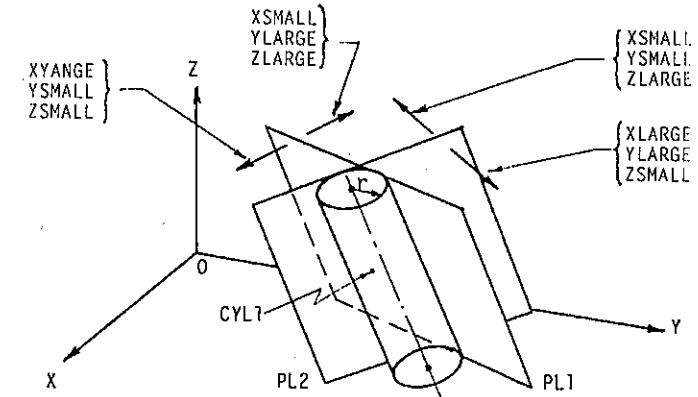
$$\left\{ \begin{matrix} P1 \\ x,y,z \end{matrix} \right\}, \left\{ \begin{matrix} V1 \\ a,b,c \end{matrix} \right\}, r$$



TABLO 4-17 DAİRESEL SİLİNDİRLERİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	Örnekler CYL1=CYLNDR/P1,V1,r CYL1=CYLNDR/x,y,z,a,b,c,r CYL1=CYLNDR/P1,a,b,c,n CYL1=CYLNDR/x,y,z,V1,r	
2	Silindir r yarıçaplı olup; verilen iki düzleme (PL1 ve PL2) teğettir.	$\text{CYLNDR} / \left\{ \begin{matrix} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \\ \text{ZLARGE} \\ \text{ZSMALL} \end{matrix} \right\}, \text{TANTO}, \text{PL1}, \$$ $\left\{ \begin{matrix} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \\ \text{ZLARGE} \\ \text{ZSMALL} \end{matrix} \right\}, \text{TANTO}, \text{PL2}, \text{RADIUS}, r$

Yorum Bir düzlem sembolünden önceki değiştirici düzeltici, tanımlanan silindirin düzlemin hangi tarafında olduğunu gösterir.



Örnekler

CYL1=CYLNDR/XLARGE TANTO,PL1,YLARGE,TANTO,PL2,RADIUS,r
CYL1=CYLNDR/ZSMALL,TANTO,PL2,ZSMALL,TANTO,PL1,RADIUS,r

- 3 Silindir r yarıçaplı olup; verilen iki nokta (P1 ve P2) ile tanımlanan doğru boyunca bir PL1 düzlemine teğettir.

$$\text{CYLNDR} / \left\{ \begin{matrix} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \\ \text{ZLARGE} \\ \text{ZSMALL} \end{matrix} \right\}, \text{TANTO}, \text{PL1}, \text{THRU}, \$$$

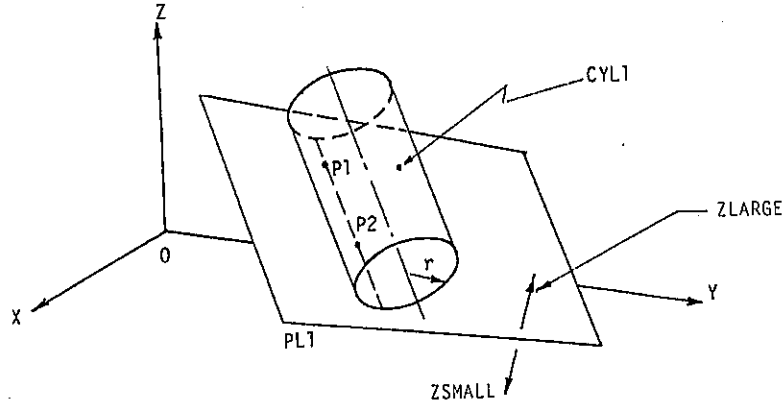
$$P1,P2,\text{RADIUS}, r$$

TABLO 4-17 DAİRESEL SİLİNDİRLERİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Şart	Deyim Formatı
-----	--------------	---------------

Örnek

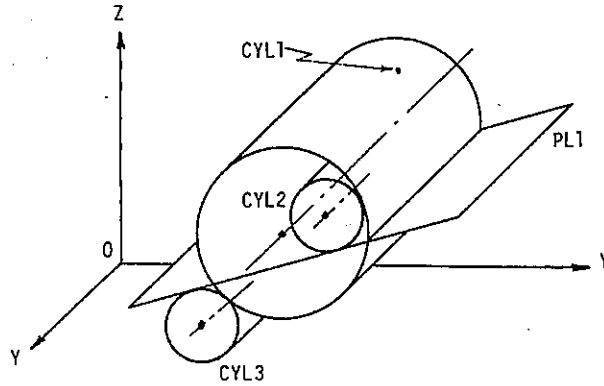
CYL1=CYLNDR/ZLARGE,TANTO,PL1,THRU,P1,P2,RADIUS,r



- 4 Silindir r yarıçaplı olup; verilen bir CYL1 silindirin ve verilen silindirin eksenine paralel olan bir PL1 düzlemine teğettir.

CYLNDR / $\left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \\ \text{ZLARGE} \\ \text{ZSMALL} \end{array} \right\}, \text{PL1}, \left\{ \begin{array}{l} \text{XLARGE} \\ \text{XSMALL} \\ \text{YLARGE} \\ \text{YSMALL} \\ \text{ZLARGE} \\ \text{ZSMALL} \end{array} \right\}, \left\{ \begin{array}{l} \text{IN} \\ \text{OUT} \end{array} \right\}, \text{CYL1}, \text{RADIUS}, r$

Yorum Bu deyim formatı, Tablo 4-10'daki No 3 daire deyim formatına benzerdir.

**Örnekler**

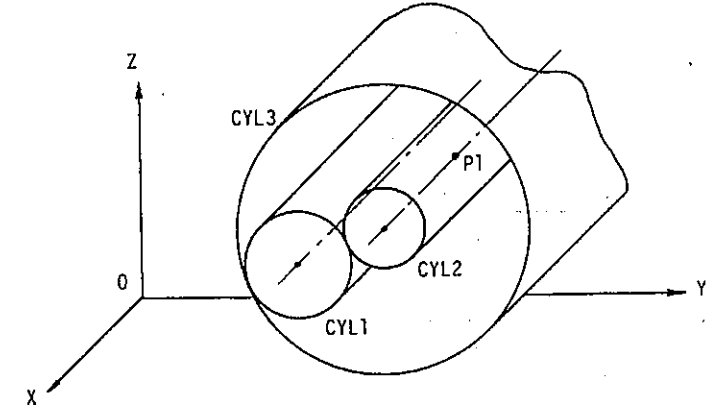
CYL2=CYLNDR/ZLARGE,PL1,ZLARGE,IN,CYL1,RADIUS,r
CYL3=CYLNDR/ZSMALL,PL1,ZSMALL,OUT,CYL1,RADIUS,r

- 5 Silindir, bir CYL1 silindirin teğet olup; eksenini verilen bir P1 noktasından geçmektedir.

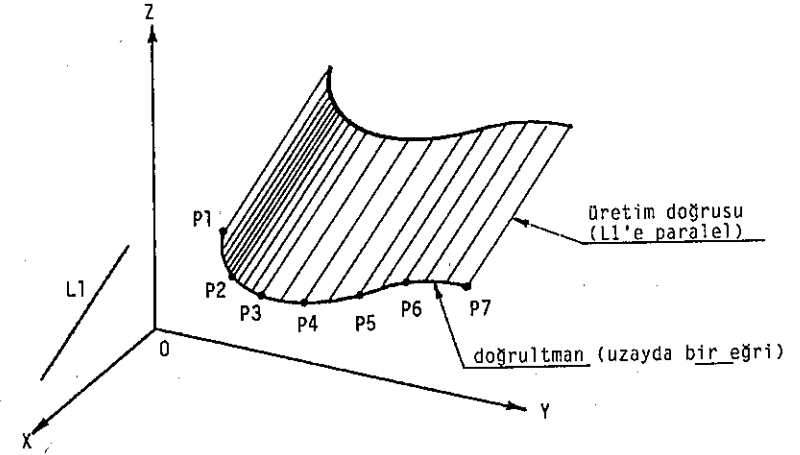
CYLNDR/CENTER,P1, $\left\{ \begin{array}{l} \text{LARGE} \\ \text{SMALL} \end{array} \right\}, \text{TANTO}, \text{CYL1}$

TABLO 4-17 DAİRESEL SİLİNDİRLERİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
-----	---------------	---------------

**Örnekler**

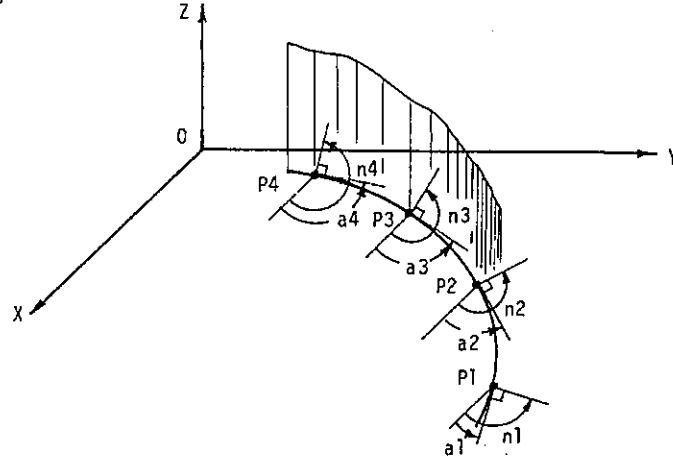
CYL2=CYLNDR/CENTER,P1,SMALL,TANTO,CYL1
CYL3=CYLNDR/CENTER,P1,LARGE,TANTO,CYL1



Şekil 4-13 Çizelgenmiş silindir

TABLO 4-18 ÇİZELGELENMİŞ BİR SİLİNDİRİ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Silindir X, Y ve Z eksenine paralel olup, doğrultmanı, $i = 1, \dots, n$ olmak üzere verilen (y_i, z_i) , (x_i, z_i) veya (x_i, y_i) noktalar kümesi, ile $i = 1, \dots, n$ olmak üzere bunlara ait "si" eğimleri veya $i = 1, \dots, n$ olmak üzere seçilmiş noktalardaki "ni" normal açıları vasıtasıyla tanımlanmaktadır. Doğrultmana teğet veya normal olan doğrunun açısı; +X (X-Y düzleminde), +Y (Y-Z düzleminde) veya +Z (Z-X düzleminde) ekseninden itibaren saat dönme yönünün ters yönünde ölçülmektedir.	TABCYL/ $\left\{ \begin{array}{l} \text{NOX} \\ \text{NOY} \\ \text{NOZ} \\ \text{RTHETA} \\ \text{THETA} \end{array} \right\} \left\{ \begin{array}{l} \text{SPLINE} \\ \text{FTSLOP} \\ \text{PTNORM} \end{array} \right\} \$$ [,TRFORM,M1], dta



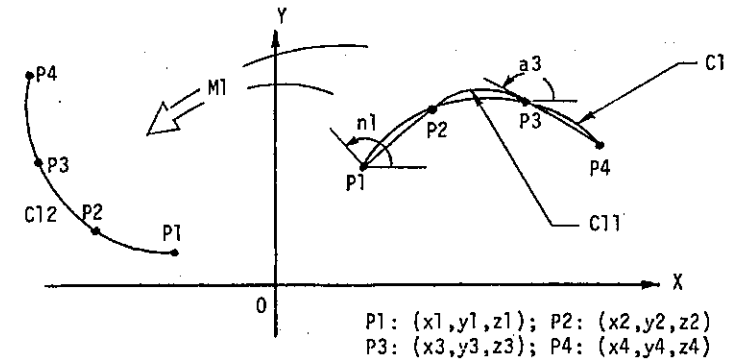
Yorum Verilen noktalar, eğimler ve normal açıları ile ilgili veriler, bu deyim'in son elemanı (dta) vasıtasıyla belirtilir (Tablo 4-19'a bakınız). İlk kelime, gerek silindirin ve gerekse verilen noktalar kümesine ait verileri ihtiva eden düzlemin yönünü belirtir. Şöyle ki:

NOX ve NOY: dta içinde listelenmiş olan verilen noktalar, sırasıyla Y-Z ve X-Z düzlemlerinde olup; sırasıyla (y_i, z_i) ve (z_i, x_i) Kartezyen koordinatlarıyla belirtilmiştir.

İkinci kelime, doğrultmanı hesaplamak ve meydana getirmek için NC işlemcisi tarafından kullanılacak yordamı belirtir. Bu sözcük tanımları aşağıdaki gibidir:

TABLO 4-18 BİR ÇİZELGELENMİŞ SİLİNDİRİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
	SPLINE: Doğrultman; $i = 1, \dots, n$ olmak üzere verilen noktaları (x_i, y_i, z_i) , verilerin seçilmiş noktalardaki eğim veya normal açısı ile birlikte ya da bu olmaksızın enterpolasyon işlemine tabi tutularak meydana getirilir. Her bir nokta (x_i, y_i, z_i) eğim(si) veya normal açısı (ni) bir veri elemanı olarak hesaba katılır. Verilen veri elemanlarının minimum sayısı üçtür (örneğin; verilen üç nokta veya verilen iki nokta ile birlikte verilen eğim ya da normal açısı). Bu seçenek, verilerin durum için en pürüzsüz eğriyi (doğrultmanı) yaratır.	
	PTSLOP: Doğrultman; verilen noktaların verilen her bir P_i noktasındaki eğim verilen si değerine eşit olacak şekilde enterpolasyon işlemine tabi tutulmasıyla meydana getirilir. En az iki nokta ve iki normal açısının dta'da belirtilmesi gerekir.	
	PTNORM: Doğrultman; verilen noktaları, verilen her bir P_i noktasındaki normal açısı verilen ni değerine eşit olmak üzere enterpolasyon işlemine tabi tutulmasıyla meydana getirilir. En az iki nokta ve iki normal açısının dta teriminde belirtilmesi gerekir.	
	TRFORM: Bu kelime, taksimatlı silindirin istenen silindire dönüştürülmesi için ihtiyaç olunan tanımlanmış bir M1 matrisli vasıtasıyla gösterilen dönüşüm işlemi (yer değiştirme, dönme veya her ikisi) belirtir.	



P1: (x_1, y_1, z_1) ; P2: (x_2, y_2, z_2)
P3: (x_3, y_3, z_3) ; P4: (x_4, y_4, z_4)

Örnekler

C1=TABCYL/NOZ,SPLINE, x 1,y 1, x 2,y 2,x 3,y 3,x 4,y 4

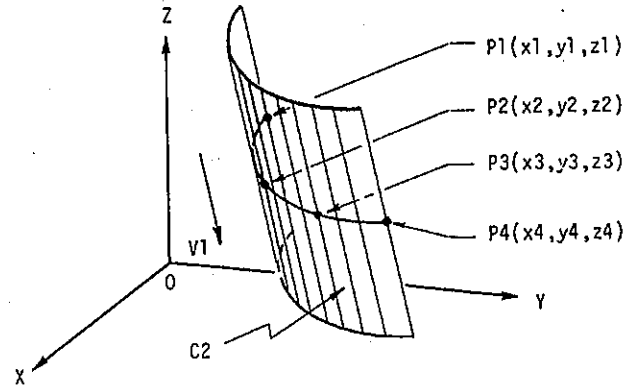
C11=TABCYL/NOZ,SPLINE,x1, y 1,NORMAL,n 1,x 2,y 2,x 3,y 3, SLOPE,s 3,x 4,y 4

C12=TABCYL/NOZ,SPLINE, TRFORM,M1,x1,y 1, x 2,y 2,x 3,y 3,x 4,y 4

İlk iki örnekte, C1 ve C11 silindirlerini tanımlamak için verilen noktalar kümesi aynen kullanılmaktadır. Bununla beraber, C11 silindirinin doğrultmanında ilave sınırlamalar (S3 eğimi ve N1 normal açısı) olduğu için, sonuçlar aynı değildir. Üçüncü örnekte ise, C1 silindirini C12 silindirine dönüştürmek için ihtiyaç olunan döndürme matrislidir.

TABLO 4-18 BİR ÇİZELGELENMİŞ SİLİNDİRİ TANIMLAYAN DEYİMLER (devam)

No.	Bilinen Durum	Deyim Formatı
2	Silindir; verilen bir V1 vektörüne paralel olup; doğrultmanı, $i = 1, \dots, n$ olmak üzere verilen bir noktalar kümesi ile seçilmiş noktalara ait s_i eğimleri veya n_i normal açıları ($i = 1, \dots, n$) vasıtasıyla tanımlanmaktadır.	TABCYL/XYZ,SPLINE, [TRFORM,M1,]V1,dta



Yorum No.1 deyim formatındaki SPLINE sözcüğüne ait açıklama geçerlidir. dta'nın içeriği:

$$x_1, y_1, z_1, \left[\begin{array}{l} \text{SLOPE}, s_1 \\ \text{NORMAL}, n_1 \end{array} \right], x_2, y_2, z_2, \left[\begin{array}{l} \text{SLOPE} \\ \text{NORMAL}, n_2 \end{array} \right]$$

$$x_n, y_n, z_n, \left[\begin{array}{l} \text{SLOPE}, s_n \\ \text{NORMAL}, n_n \end{array} \right]$$

Örnek

C2=TABCYL/XYZ,SPLINE,V1,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4

Giriş noktaları koordinat düzlemlerinde belirtilmiş olarak TABCYL deyimini tarafından tanımlanan bir çizelgelenmiş silindir şu karakteristik özelliklere sahiptir:

1. Meydana getirilen doğrultman, deyimde belirtilen iki komşu noktayı birleştiren bir dizi kübik polinomial eğrilerden oluşur. NC işlemcideki hesaplama (veya eğri uydurma) yordamı, belirtilen noktaların tümünden geçen pürüzsüz bir eğrinin meydana getirilmesini sağlamak üzere, belirtilen bir noktadaki iki komşu eğrinin eğimlerinin aynı olmasını sağlar. Eğer eğimler veya normal açıları deyimde belirtilmişse; bunlar, kübik polinomial

eğrilerin hesaplanması için eğri uydurma yordamları tarafından kullanılacaktır.

2. Eğer belirtilen ilk ve son noktadaki eğimler deyimde belirtilmişse; bu eğimler, belirtilen noktalardan geçen pürüzsüz bir eğri elde edilebilecek biçimde belirlenir. Aksi takdirde, belirtilen eğimle ve normal açıları kullanılır.
3. Meydana getirilen doğrultman, TABCYL deyiminde belirtilen ilk ve son noktalarda sona ermeyip, 10 birim kadar doğru biçimde bu noktaların dışına uzanır. Bu uzatılmış doğruların eğimleri, sırasıyla iki uç noktasındaki doğrultma eğimleri ile aynıdır.

Bir TABCYL deyimini işledikten sonra, NC işlemcisi, doğrultmanı kontrol etmek, düzeltmek veya değiştirmek için ihtiyaç duyulan bilgileri yazar. Yazıcı çıktısının içeriği ve biçimi Ek C'de verilmiştir.

CYLNDR ve TABCYL deyimleri tarafından tanımlanan silindirik yüzeyler sonsuz uzunluktadır.

Bir döner yüzey, bir düzlem eğrisinin, eğriyi ihtiva eden düzlemde bulunan sabit bir doğru veya eksen etrafında dönmesiyle süpürülen yüzeydir. Dairesel silindirler, koniler, küreler ve torlar buna örnek teşkil eder. Konileri ve küreleri tanımlayan deyimler, sırasıyla:

CONE/.....

ve

SPHERE/.....

biçimindedir. Deyim formatları ve bunların kullanılışı ile ilgili ayrıntılı bilgi, Tablo 4-20 ve 4-21'de verilmiştir. Bir tor (Şekil 4-14)

TORUS/P1,V1,r1,r2

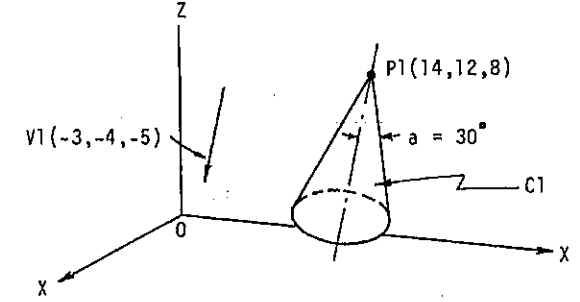
deyimile tanımlanabilir. Bu deyimde, P1 torun merkezi; V1, torun eksenine paralel olan bir vektör; r1, torun merkezinden toru meydana getiren dairenin merkezine olan mesafe; r2 ise toru meydana getiren dairenin yarıçapıdır.

TABLO 4-19 TABCYL DEYİMİNDEKİ DATA'NIN İÇERİĞİ

Kelime	Düzlemde verilen noktalar	SPLINE	PTSLOP	PTNORM
NOZ	X-Y	$x1, y1, x2, y2, \dots, xn, yn$	$x1, y1, s1, x2, y2, s2, \dots, xn, yn, sn$	$x1, y1, n1, x2, y2, n2, \dots, xn, yn, nn$
RTHETA		$r1, a1, r2, a2, \dots, rn, an$	$r1, a1, s1, r2, a2, s2, \dots, rn, an, sn$	$r1, a1, n1, r2, a2, n2, \dots, rn, an, nn$
THETAR		$a1, r1, a2, r2, \dots, an, rn$	$a1, r1, s1, a2, r2, s2, \dots, an, rn, sn$	$a1, r1, n1, a2, r2, n2, \dots, an, rn, nn$
NOX	Y-Z	$y1, z1, y2, z2, \dots, yn, zn$	$y1, z1, s1, y2, z2, s2, \dots, yn, zn, sn$	$y1, z1, n1, y2, z2, n2, \dots, yn, zn, nn$
NOY	Z-X	$z1, x1, z2, x2, \dots, zn, xn$	$z1, x1, s1, z2, x2, s2, \dots, zn, xn, sn$	$z1, x1, n1, z2, x2, n2, \dots, zn, xn, nn$
Açıklamalar		<p>1. Gerekliği takdirde, i noktasının koordinatlarından sonra, önce SLOPE veya NORMAL kelimesi yazılıp, yanına i noktasındaki eğimi veya ni normal açısı belirtilebilir. Örneğin: $x1, y1, x2, y2, SLOPE, s2, \dots, xn, yn$ veya $x1, y1, NORMAL, n1, x2, y2, SLOPE, s2, \dots, xn, yn$</p> <p>2. En az nokta sayısı 3'tür.</p> <p>3. Bu seçenek, verilen noktalar kümesi için en uygun eğriyi meydana getirir.</p>		

TABLO 4-20 BİR KONİYİ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Koninin tepe noktası verilen bir P1(x,y,z) noktasında ve tepe açısının yarısı olup; koni eksenini bir V1(i,j,k) vektörüne paraleldir.	$CONE / \{CANON, x, y, z, i, j, k, \cos(a)\}$ $P1, V1, a$



Yorum Birinci seçenek, NC işlecisinin tanımlanmış olan bir koninin verilerini saklayabildiği standart (kanonik) biçimdir. i, j ve k değerlerinin, sırasıyla bir V1 birim vektörünün X, Y ve Z bileşenleri olması gerekir. İkinci seçenekte ise, V1, birim vektörün dışında bir vektör olabilir.

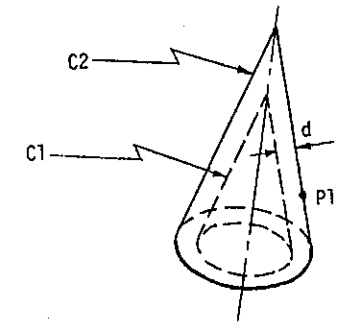
Örnekler

$C1 = CONE/CANON, 14, 12, 8, -12, 8, -0.4243, -0.5657, -0.7071, 0.8660$
 $C2 = CONE/P1, V1, 30$

- 2 Koni, verilen bir C1 konisi ile eşeksenli olup; bu verilen koniden bir d uzaklıktadır.

$CONE/PARLEL, C1, STEP, \left\{ \begin{matrix} IN \\ OUT \end{matrix} \right\}, d$

Yorum IN ve OUT deęiřtiricileri, tanımlanan koninin verilen C1 konisinin içinde veya dışında olduğunu gösterir. d deęeri, iki yüzey arasındaki uzaklıktır.

**Örnekler**

$C2 = CONE/PARLEL, C1, STEP, OUT, d$
 $C1 = CONE/PARLEL, C2, STEP, IN, d$

- 3 Koni, verilen bir C1 konisi eşeksenli olup; verilen bir P1 noktasından geçmektedir.

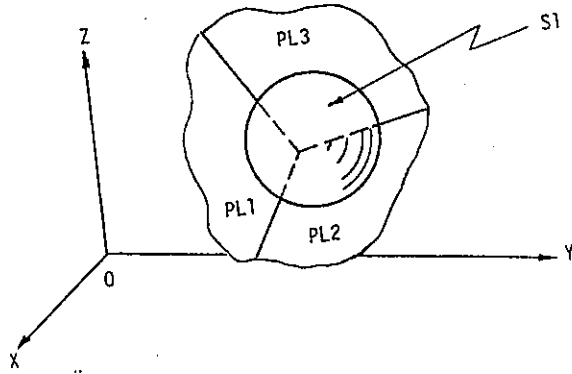
$CONE/PARLEL, C1, THRU, P1$

Örnek

$C2 = CONE/PARLEL, C1, THRU, P1$

TABLO 4-21 KÜRELERİ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Kürenin merkezi verilen bir P1 (x,y,z) noktasında olup, yarıçapı r'dir.	SPHERE / $\left\{ \begin{array}{l} x,y,z,r \\ P1,r \\ CENTER,P1,RADIUS,r \end{array} \right\}$
2	Küre; merkezi verilen bir P1 noktasında olup, verilen değer bir P2 noktasından geçmektedir.	SPHERE/CENTER,P1,P2
3	Küre; koordinatları verilen bir P1 merkezli olup, verilen bir PL1 düzlemine teğettir.	SPHERE/CENTER,P1,TANTO,PL1
4	Küre; verilen dört noktadan (P1,P2,P3 ve P4) geçmektedir.	SPHERE/P1,P2,P3,P4
5	Küre; verilen bir r yarıçapına sahip olup; verilen üç düzleme (PL1,PL2 ve PL3) teğettir.	SPHERE / $\left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \\ ZLARGE \\ ZSMALL \end{array} \right\} ,PL1, \left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \\ ZLARGE \\ ZSMALL \end{array} \right\} ,\$$ $\left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \\ ZLARGE \\ ZSMALL \end{array} \right\} ,PL2, \left\{ \begin{array}{l} XLARGE \\ XSMALL \\ YLARGE \\ YSMALL \\ ZLARGE \\ ZSMALL \end{array} \right\} ,PL3,RADIUS,r$



Örnek

S1 = SPHERE/YLARGE,PL1,XLARGE,PL3,ZLARGE,PL2,RADIUS,5

4.8 SINIRLANDIRILMIŞ (RULED) YÜZEYLERİ TANIMLAYAN DEYİMLER

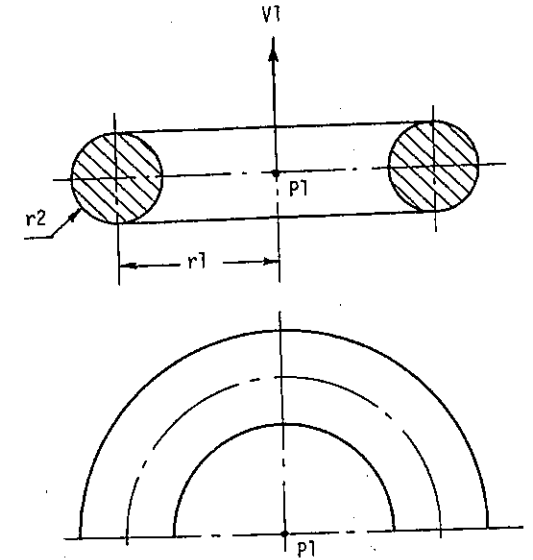
Mühendislik tasarımında, genellikle bir kesitten diğerine yumuşak geçiş yapan bir yüzeye ihtiyaç duyarız. Bununla beraber, iki kesitin biçimleri, Şekil 4-15'te görülebileceği gibi tamamen farklı olabilir. Bu yumuşak geçiş yüzeyini tasarlamak için uyguladığımız yöntemlerden birisi, taşıma yöntemi denilen ve iki

kesitin profilleri olan iki düzlemsel eğri üzerinde karşılık gelen noktaları doğru çizgiler vasıtasıyla birleştirmektir (Şekil 4-16). İki düzlemsel eğri üzerinde karşılık gelen noktaları bulabilmek için, sırasıyla aşağıdaki işlemleri yapınız:

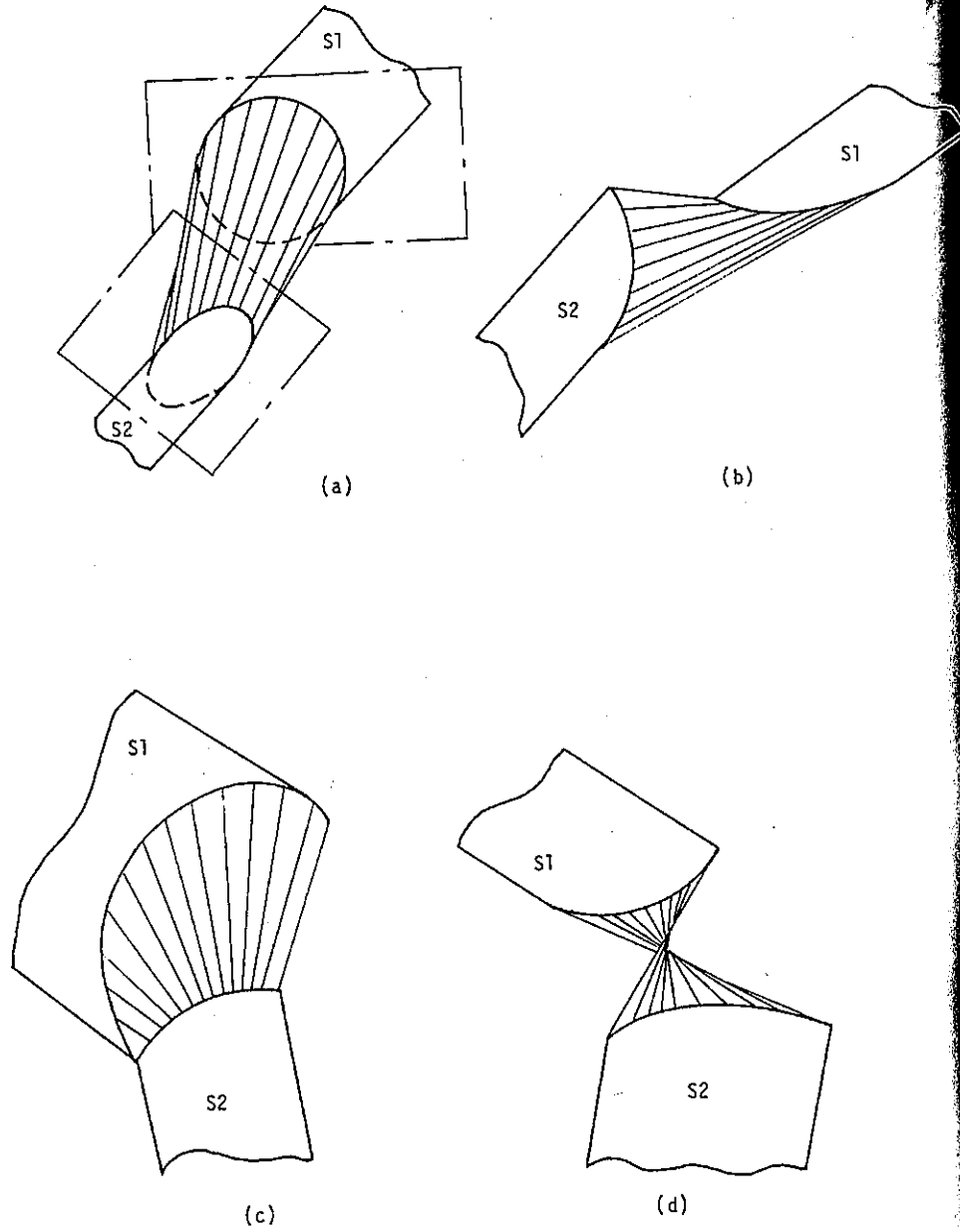
1. Herbir eğri üzerinde başlangıç ve bitiş noktalarını birleştiren bir taban doğrusu ve kirişi çiziniz.
2. İlgili başlangıç noktalarından itibaren aynı yüzde mesafesine sahip olmak üzere, iki taban doğrusu üzerinde n ve m noktalarını yerleştiriniz.
3. n ve m noktalarında taban doğrularına dik doğrular çiziniz. Bu iki doğrunun iki eğri ile kesiştiği noktalar, karşılık gelen p ve q noktalarıdır.

Bu yöntemle oluşturulan bir yüzeye, APT dilinde "sınırlandırılmış yüzey" denir. Yukarıda verilen açıklamadan açıkça anlaşılacağı gibi, iki düzlemsel eğrinin biçimi bakımından kısıtlamalar vardır. Belli bir kısmı, başlangıç ve bitiş noktaları tarafından tanımlanan aralığın dışında bulunan veya taban doğrusuna dik olan doğru ile birden fazla kesim noktasına sahip bir düzlemsel eğriden kaçınılmalıdır (Şekil 4-17'de D alanına bakınız). Eğer bu tür bir eğri esasına dayalı bir sınırlandırılmış yüzey oluşturmak zorunlu ise; bu eğrinin, her biri taban doğrusuna dik doğru ile yalnızca bir kesim noktasına olmak üzere, iki veya daha çok parçaya bölünmesi gerekir (Şekil 4-18'de C1 ve C2 eğrilerine bakınız). Diğer düzlemsel eğri de, tasarım gereksinimine göre, aynı sayıda parçaya (C3 ve C4) bölünmelidir.

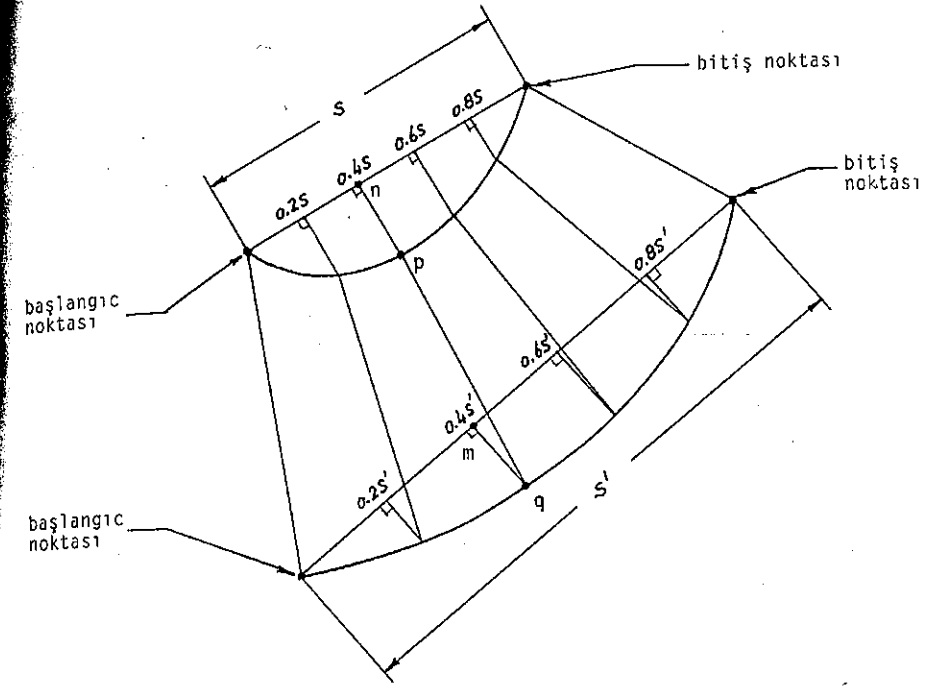
Bir sınırlandırılmış yüzeyi tanımlamak için kullanılan iki düzlemsel eğri, genellikle iki yüzeyin kesitleri, yani iki yüzeyin iki düzlemle kesişimleridir (Şekil 4-15[a]). Dolayısıyla, APT dilinde taşınmış yüzeylerin tanımlanması için ihtiyaç olunan elemanlar; iki yüzey, iki yüzeyi kesen iki düzlem ve iki düzlemsel eğrinin



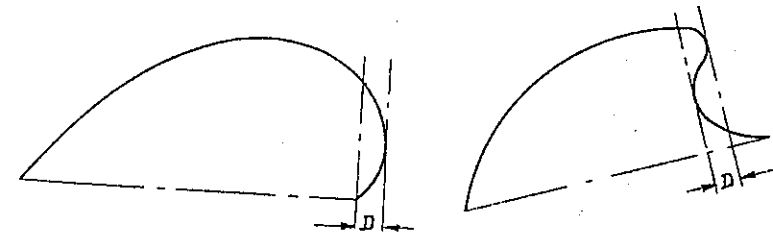
Şekil 4-14 Bir tor ve parametreleri



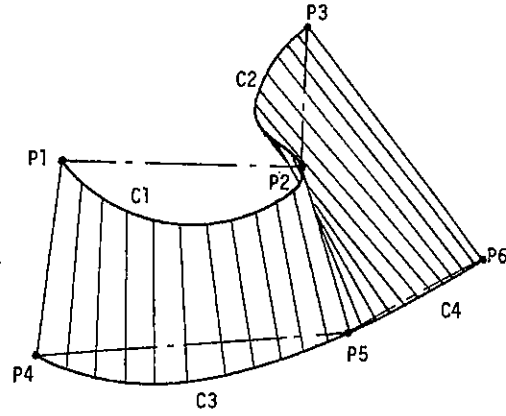
Şekil 4-15 Geçiş yüzeyleri



Şekil 4-16 Bir sınırlandırılmış yüzeyin yapısı



Şekil 4-17 Bir sınırlandırılmış yüzeyi oluşturmak için düzensel eğriler kullanılmamalıdır.



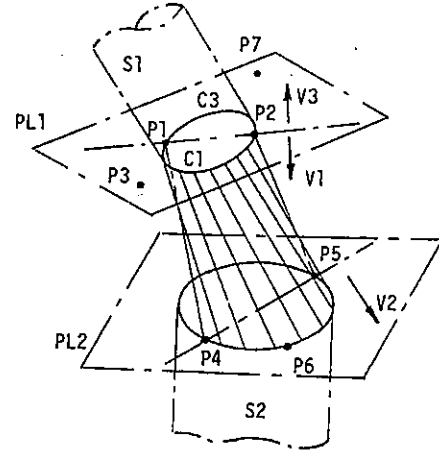
Şekil 4-18 Üstteki düzlemsel eğrinin taban doğrusuna dik doğru ile iki kesim noktası vardır. Sınırlanmış yüzey, düzlemsel eğrilerin her birini iki bölüme ayırmak suretiyle oluşturulur.

başlangıç ve bitiş noktalarıdır. Bir sınırlanmış yüzeyi tanımlayan deyim RLDSRF/.....

olup, bu deyimin formatları ve kullanılışı Tablo 4-22'de sırasıyla belirtilmiştir.

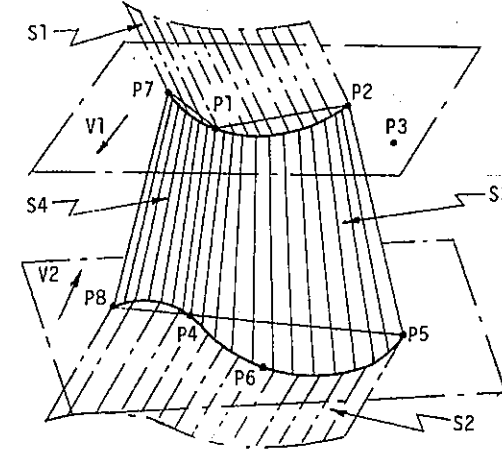
TABLO 4-22 SINIRLANDIRILMIŞ YÜZEYLERİ TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	İki düzlemsel eğri, verilen iki yüzeyin (S1 ve S2) sırasıyla iki düzlemle (PL1 ve PL2) arakesitidir. PL1 düzlemi, düzlemsel eğrinin verilen başlangıç ve bitiş noktaları (P1 ve P2) ve diğer bir nokta (P3) veya bir V1 vektörü ile tanımlanmakta; PL2 düzlemi de, düzlemsel eğrinin verilen başlangıç ve bitiş noktaları (P4 ve P5) ve diğer bir nokta (P6) veya vektör (V2) ile tanımlanmaktadır. P3 noktası, PL1 düzlemindeki taban doğrusu üzerindeki dışarda herhangi bir nokta olabilir; P6 noktası ise, PL2 düzlemindeki taban doğrusu üzerindeki dışarda herhangi bir nokta olabilir.	$RLDSRF/S1,P1,P2,\left\{\begin{matrix} P3 \\ V1 \end{matrix}\right\},S2,P4,P5,\left\{\begin{matrix} P6 \\ V2 \end{matrix}\right\}$



TABLO 4-22 SINIRLANDIRILMIŞ YÜZEYLERİ TANIMLAYAN DEYİMLER (devam)

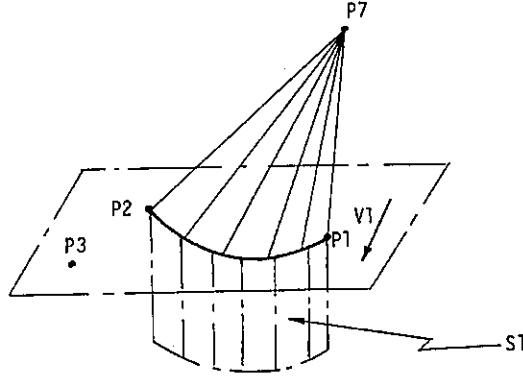
No.	Bilinen Durum	Deyim Formatı
-----	---------------	---------------



Yorum S1 yüzeyini izleyen elemanlar (yani P1, P2 ve P3 noktaları veya V1 vektörü) aynı PL1 düzleminde olmalı; S2'yi izleyenler ise PL2 düzleminde bulunmalıdır. Bir eğrinin başlangıç ve bitiş noktalarını birleştiren taban doğrusu, eğriyi iki kısma böler. Yüzeyi izleyen üçüncü nokta, sınırlanmış yüzeyin sınır olarak seçilecek kısmı gösterir. Örneğin, eğer P3 noktası veya V1 vektörü belirtilmişse, taban doğrusunun aynı tarafındaki kısım (yani C1 eğrisi) seçilir. Bunun tersi olarak, eğer P7 noktası veya V3 vektörü belirtilmişse, C3 eğrisi seçilir. Bu nedenle, deyimdeki P3 ve P6 noktaları taban doğruları üzerinde olmamalıdır. Aynı sebepten dolayı, V1 ve V2 vektörleri de sırasıyla P1P2 ve P4P5 taban doğrularına paralel olmamalı; taban doğrusu ile küçük bir açı yapan (örneğin 5 dereceden daha az) bir doğrultuda da bulunmamalıdır. Bir düzlemsel eğri, başlangıç ve bitiş noktalarının dışında, taban doğrusu ile hiçbir kesişime sahip olmamalı; aksi takdirde, kısımlara bölünmüş olmalıdır.

TABLO 4-22 SINIRLANDIRILMIŞ YÜZEYLERİ TANIMLAYAN DEYİMLER (Devam)

No.	Bilinen Durum	Deyim Formatı
Örnekler		
	S3 = RLDSRF/S1,P1,P2,P3,S2,P4,P5,P6	
	S4 = RLDSRF/S1,P1,P7,V1,S2,P4,P8,V2	
2	Yukarıdaki formatta belirtilen yüzeylerin biri, verilen bir P7 noktasıdır.	RLDSRF/S1,P1,P2, $\left\{ \begin{matrix} P3 \\ V1 \end{matrix} \right\}, P7$



4.9 ÇOK KONİLİ BİR YÜZEYİ TANIMLAYAN DEYİMLER

Mühendislik tasarımında, karmaşık şekilli bir parçanın profili, genellikle bir dizi kesit eğrileri esasına dayalı olarak tanımlanır. Boylamasına doğrultu boyunca uzanan profil ise, kesit eğrileri üzerinde karşılık gelen noktaları polinom fonksiyonlarıyla tanımlanan doğru veya eğrilerle birleştirmek suretiyle belirlenebilir.

APT dilinde bir çok konili yüzey, yüzeyin boylamasına eksenine dik durumdaki kesitlerinin yaklaşık olarak koni biçiminde olduğu varsayılan ve boylamasına eksen boyunca profilinin polinom fonksiyonlarıyla tanımlandığı üç boyutlu bir yüzeydir. Örneğin Şekil 4-19'da gösterilen kanada benzer yüzeyin, yaklaşık olarak bir çok konili yüzey olduğu varsayılır. Bu parçanın boyuna eksenini, X eksenine paraleldir. Parçanın, örneğin $x = x'$ noktasında, keyfi olarak alınan bir kesitin bir konik eğri olarak tanımlanabilmesi için, iki kenarının sırasıyla kesit eğrisinin iki uç noktasındaki (P1 ve P2) teğet doğruları olduğu bir P1-P4-P2-P3 paralel kenarını oluşturmak gerekir. Bir kesit eğrisini tanımlamak için şu yedi parametreye ihtiyaç vardır: P1 noktasının L ve H koordinatları P1 noktasından P2 noktasına ve P3 noktasından P2 noktasına kadar olan artışı koordinatlar (yani; C, B, A ve D), kesit eğrisinin P3-P4 köşegeniyle kesişimini belirleyen $K = L2/L1$ oranı.*

*Tanımlanmış olan çok konili yüzey kesit eğrisinin köşegenle gerçek kesişim noktası, K değeri tarafından belirtilen kesişim noktasının tam olarak aynı olmayabilir. K parametresi, kesit eğrisinin yaklaşık biçimini tanımlamak için kullanılır. Bu konuda ayrıntılı açıklama için Problem 8.10'a bakınız.

Kesitin biçimi X koordinatı ile değişir; dolayısıyla, bu yedi parametre de x 'in fonksiyonlarıdır. Bu $A(x)$, $B(x)$, $C(x)$, $D(x)$, $L(x)$, $H(x)$, ve $K(x)$ değişimleri, aşağıdaki polinom fonksiyonları vasıtasıyla tanımlanabilir:

$$\left. \begin{aligned} A(x) &= P_{A0}' + P_{Aq}' x^{\frac{1}{2}} + P_{A1}' x + P_{A2}' x^2 + P_{A3}' x^3 + P_{A4}' x^4 + P_{A5}' x^5 \\ &\quad + P_{A6}' x^6 + P_{A7}' x^7 \\ \dots \\ K(x) &= P_{K0}' + P_{Kq}' x^{\frac{1}{2}} + P_{K1}' x + P_{K2}' x^2 + P_{K3}' x^3 + P_{K4}' x^4 + P_{K5}' x^5 \\ &\quad + P_{K6}' x^6 + P_{K7}' x^7 \end{aligned} \right\}$$

Burada x ; $x1$ ve $x2$ arasında değişir (yani; $x \in [x1, x2]$ 'dir). Matematiksel gösterimin basite indirgenebilmesi bakımından, x değişkenini, x yerine fonksiyonu konularak normalleştirmek daha uygundur.

$$x = t \cdot (x2 - x1) + x1$$

Burada; $t \in [0, 1]$ 'dir. Böylece, yukarıda sıra ile belirtilmiş olan polinom fonksiyonları aşağıdaki biçimde basitleştirilebilir:

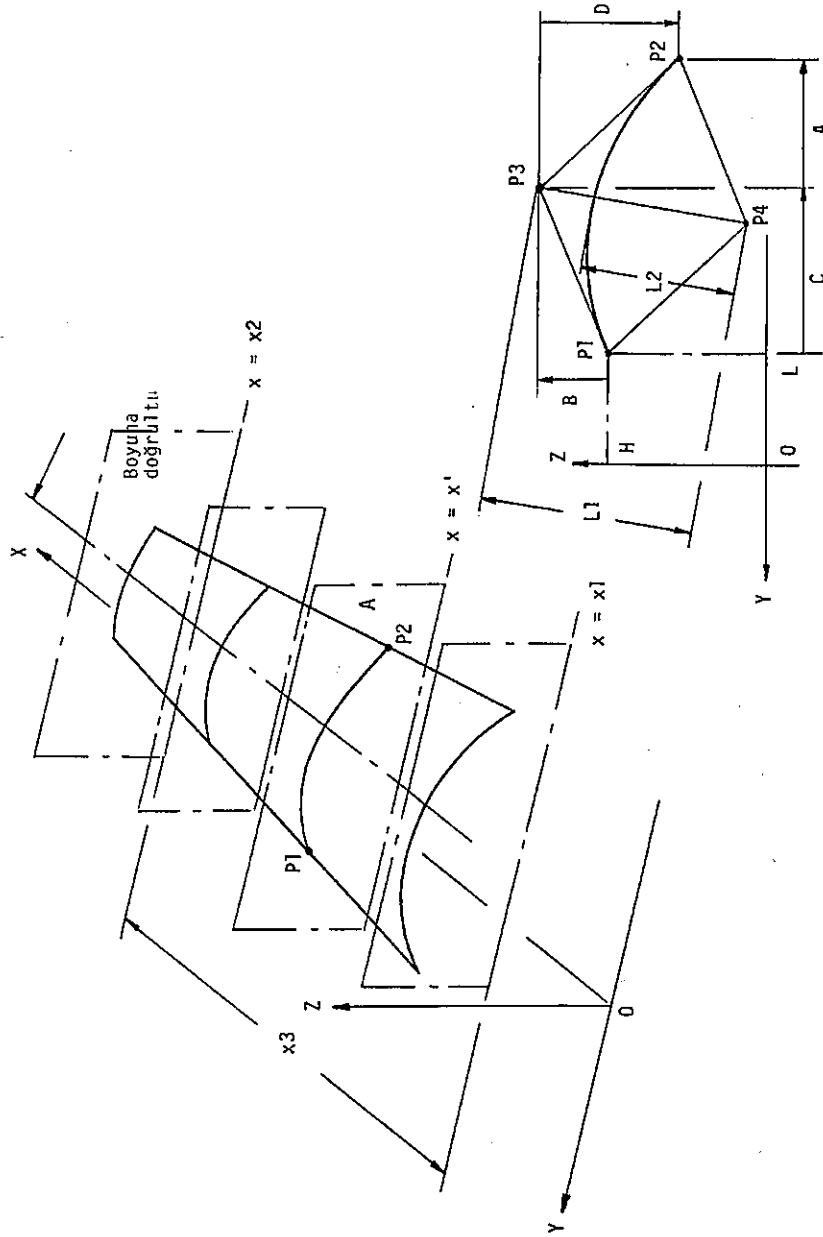
$$\left. \begin{aligned} A(t) &= P_{A0} + P_{Aq} t^{\frac{1}{2}} + P_{A1} t + P_{A2} t^2 + P_{A3} t^3 + P_{A4} t^4 + P_{A5} t^5 \\ &\quad + P_{A6} t^6 + P_{A7} t^7 \\ \dots \\ K(t) &= P_{K0} + P_{Kq} t^{\frac{1}{2}} + P_{K1} t + P_{K2} t^2 + P_{K3} t^3 + P_{K4} t^4 + P_{K5} t^5 \\ &\quad + P_{K6} t^6 + P_{K7} t^7 \end{aligned} \right\}$$

Burada; $t \in [0, 1]$ 'dir.

APT dilinde, bir çok konili yüzey, aşağıdaki deyim tarafından tanımlanır (Şekil 4-19):

PLC1=POLCON/CANON, j,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11, m12,t1,x1,x3,\$

PA0,PA1,PA2,PA3,PA4,PA5,PA6,PA7, \$
 PB0,PB1,PB2,PB3,PB4,PB5,PB6,PB7, \$
 PC0,PC1,PC2,PC3,PC4,PC5,PC6,PC7, \$
 PD0,PD1,PD2,PD3,PD4,PD5,PD6,PD7 \$
 PL0,PL1,PL2,PL3,PL4,PL5,PL6,PL7 \$
 PH0,PH1,PH2,PH3,PH4,PH5,PH6,PH7 \$
 PK0,PK1,PK2,PK3,PK4,PK5,PK6,PK7 \$
 PAQ,PBQ,PCQ,PDQ,PLQ,PHQ,PKQ



Şekil 4-19 Boyuna doğrultusu X eksenine paralel olan bir kanada benzer yüzey.
A_KESİT DÜZLEMİ

Burada;

PLC1 = Tanımlanan yüzey sembolü

PAQ = PBQ = ... = PKQ = 0 ise, j = 1

Bunun dışında, = 2

m_1, \dots, m_{12} = Çok konili yüzeyin gerçek konum ve yönünü, Şekil 4-19'da gösterildiği gibi, yüzeyin tanımlanması için uygun konum ve yöne dönüştüren, aşağıdaki dönüşüm matrisinin 12 katsayısıdır.*

$$\begin{bmatrix} m_1 & m_2 & m_3 & m_4 \\ m_5 & m_6 & m_7 & m_8 \\ m_9 & m_{10} & m_{11} & m_{12} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Eğer parçanın boyuna eksenini X eksenine paralel ise, dönüşüm işlemine gerek yoktur. Böyle bir durumda; m_1, m_6 ve m_{11} 'in değeri 1'e eşit; diğerleri ise sıfırdır. Ayrıca; t_1 = Yüzeyde bırakılan işleme payı; x_1 = ilk kesitin x koordinatı; x_3 = ilk ve son kesitler arasındaki ve X doğrultusundaki uzaklık ve; $PA_0, PA_1, \dots, PKQ = 4.5$ nolu eşitlikteki katsayılarıdır. Bu katsayılar, boyuna doğrultunun X eksenine paralel olmasını sağlayacak şekilde belirlenir. Parçanın yukarıda belirtilen uygun yönde olmaması durumunda ise, bu katsayılar, parça m_1 'den m_{12} 'ye kadar (dahil) olan katsayılarla tanımlanan dönüşüm matrisi tarafından istenen konum ve yöne dönüştürülmüş gibi belirlenir.

Örnek

Şekil 4-20'de gösterilen PLC1 yüzeyinin, bir çok konili yüzey olarak tanımlanması gerekmektedir. Yüzey, önce, M1 dönüşüm matrisi vasıtasıyla normalleştirilmiş bir konum ve yöne dönüştürülür (yani; boyuna eksen X eksenine paralel konuma getirilir). 4.5 nolu eşitlikteki yedi polinom fonksiyonunun katsayılarını belirlemek için, C1, C2 ve C3 doğrularının t-Z ve t-Y düzlemleri üzerinde izdüşümlerini almamız gerekir. Buna göre ortaya çıkan eğriler, sırasıyla $C_{1z}, C_{1y}, C_{2z}, C_{2y}, C_{3z}$ ve C_{3y} 'dir. Bu eğrilerin, şekilde gösterildiği gibi olduğunu varsayalım. Bu duruma göre ortaya çıkan fonksiyonlar, (Şekil 4-19'a da bakınız), aşağıdaki gibidir:

$$L(t) = C_{1y}(t) = 1 - 0.2t$$

$$H(t) = C_{1z}(t) = 1.7 - 0.2t + 0.1t$$

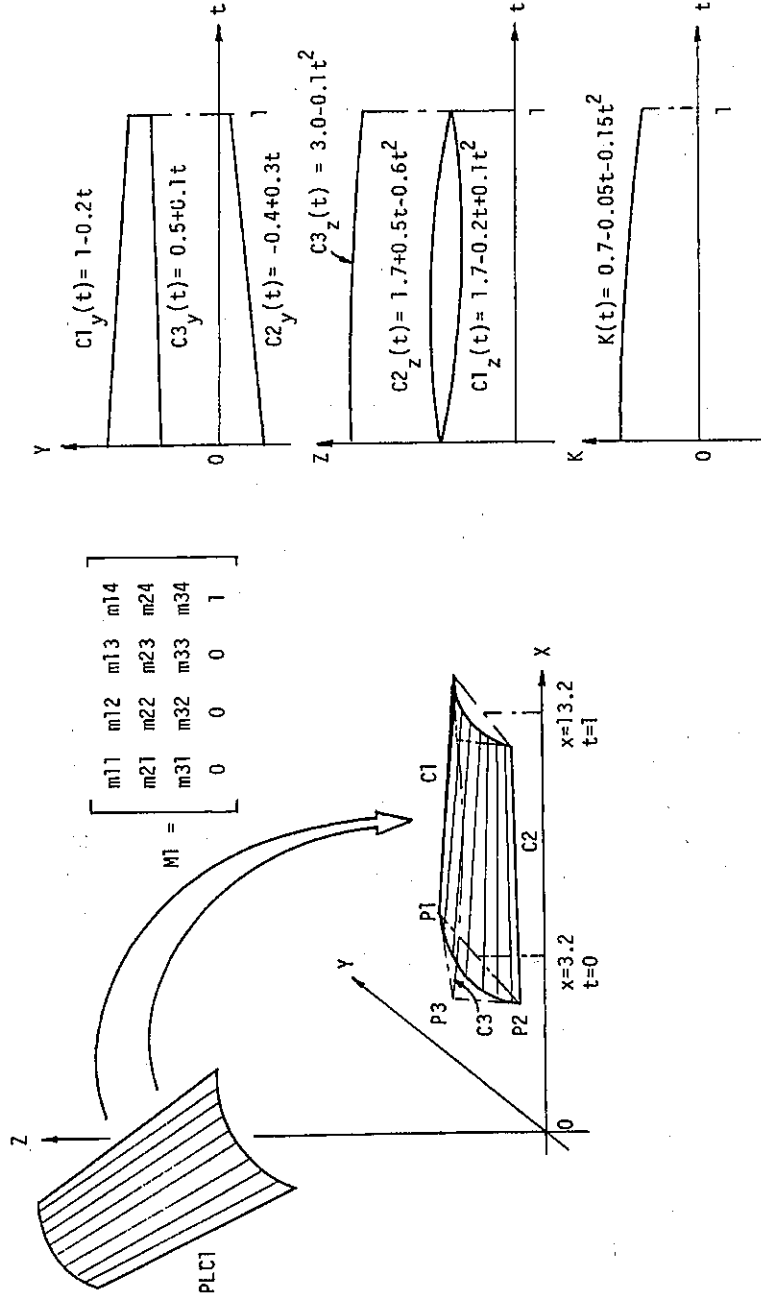
$$A(t) = C_{2y}(t) - C_{3y}(t) = -0.4 + 0.3t - (0.5 + 0.1t)$$

$$= -0.9 + 0.2t$$

$$B(t) = C_{3z}(t) - C_{1z}(t) = 3.0 - 0.1t^2 - (1.7 - 0.2t + 0.1t^2)$$

$$= 1.3 + 0.2t$$

* Bölüm 7.6'ya bakınız.



Şekil 4-20 POLCON deyimiiyle tanımlanacak bir yüzey. Parçanın gerçek konum ve yönü PLC1'dedir. Parçanın uygun bir konum ve yöne dönüştürülmesi, M1 dönüşüm matrisi vasıtasıyla sağlanır. Daha sonra C1, C2 ve C3 doğruların t-Y ve t-Z düzlemleri üzerindeki izdüşümleri elde edilir. Buna göre ortaya çıkan eğriler, sırasıyla C1y, C1z, C2y, C2z, C3y ve C3z'dir.

$$\begin{aligned} C(t) &= C3_y(t) - C1_y(t) = 0.5 + 0.1t - (1 - 0.2t) = -0.5 + 0.3t \\ D(t) &= C2_z(t) - C3_z(t) = 1.7 + 0.5t - 0.6t^2 - (3.0 - 0.1t^2) \\ &= -1.3 + 0.5t - 0.5t^2 \\ K(t) &= 0.7 - 0.05t - 0.15t^2 \end{aligned}$$

Bu ifadelerde kare kök terimlerinin hepsi sıfır olduğundan, j parametresi 1 değerine sahiptir. Ayrıca, Şekil 4-20'den görüleceği gibi, $x_1 = 3.2$ ve $x_3 = 10$ 'dur.

Dolayısıyla, bu çok konili yüzeyin, tezgaha işleme payı sıfır olacak şekilde tanımlanması için kullanılacak deyim aşağıdaki gibidir:

```
PLC1=POLCON/CANON, 1,m11,m12,m13,m14,m21,m23,m24,m31,m32,m33,m34,$
-0.3,2,10,$
-0.9,0.2,0,0,0,0,0,0,$
1.3,0.2,0,0,0,0,0,0,$
-0.5,0.3,0,0,0,0,0,0,$
-1.3,0.5,-0.5,0,0,0,0,0,$
1,0.2,0,0,0,0,0,0,$
1.7,-0.2,0.1,0,0,0,0,0,$
0.7,-0.05,-0.15,0,0,0,0,0,$
0,0,0,0,0,0,0
```

4.10 NOKTALARIN ZSURF DEYİMİYLE TANIMLANMASI

X-Y düzlemine paralel olmayan bir düzlemde yerleşik bulunan çok sayıda noktayı tanımlama gereği ile sık olarak karşılaşılır (Şekil 4-21). Teknik resimde, bu noktaların ölçüleri normal olarak X-Y düzleminde belirtilir; z koordinatları ise açık olarak gösterilmez ve bu nedenle de hesaplanması gerekir. Böyle bir durumda, çok sıkıcı ve yorucu hesap işlemlerini ortadan kaldırmak için, (yalnızca söz konusu noktaların X ve Y koordinatlarını içeren POINT deyimleri ile birlikte) ZSURF deyiminden yararlanabiliriz.

APT programı

```

.....
.....
PL0=PLANE/0,0,1,3
PL1=PLANE/2,1,3,2,3,3,4,1,2
PL2=PLANE/4,1,2,4,3,2,5,1,1
ZSURF/PL1
P0=POINT/1.5,2.0,3.0
P1=POINT/2.5,2.5
P2=POINT/3.5,2.5
P3=POINT/2.5,1.5
P4=POINT/3.5,1.5
ZSURF/PL2
P5=POINT/4.3,2.0
P6=POINT/4.7,2.0
ZSURF/4.0
P00=POINT/1.5,2.0
P11=POINT/2.5,2.5
P21=POINT/3.5,2.5
P31=POINT/2.5,1.5
P41=POINT/3.5,1.5
P51=POINT/4.3,2.0
P61=POINT/4.7,2.0
.....
.....

```

Şekil 4-21 Noktaların ZSURF deyimine tanımlanması.

$$ZSURF \left\{ \begin{array}{l} PL1 \\ a, b, c, d \\ e \end{array} \right\}$$

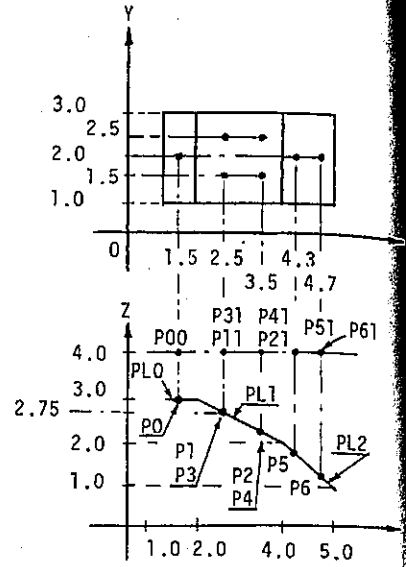
olup burada; PL1, tanımlanacak noktaların yerleşik bulunduğu düzlem; a,b,c ve d, düzlemi tanımlayan

$$ax + by + cz = d$$

eşitliğinin parametreleri; e ise, düzlemin X-Y düzlemine paralel olması halinde z koordinatıdır.

ZSURF deyimini belirtildiği takdirde, ZSURF deyimine tanımlanan düzlemde bulunan noktalar da sadece x ve y koordinatlarıyla (yani, X-Y düzlemi üzerindeki izdüşümleriyle) tanımlanabilir. NC işlemcisi, X-Y düzleminde tanımlanan noktalardan ZSURF deyimini vasıtasıyla tanımlanan düzleme izdüşüm doğruları oluşturmak suretiyle Z koordinatlarını hesaplar. Eğer ZSURF deyimini belirtilmemiş ise,

ZSURF/0



biçimi geçerlidir (yani düzlem, X-Y düzlemidir). ZSURF deyimini, diğer bir ZSURF deyimini tarafından iptal edilinceye kadar etkisini sürdürür.

Şekil 4-21, ZSURF deyiminin anlam ve kullanımını açıklayan bir örneği göstermektedir. Listelenmiş APT programından görülebileceği gibi, P1'den P6'ya kadar (dahil) olan noktalar ile P00'dan P61'e kadar (dahil) olan noktalar x ve y koordinatlarıyla tanımlanmakta olup; bunlara ait ZSURF deyimleri, müteakip POINT deyimlerinde tanımlanan noktaların z koordinatlarını hesaplamak için gereken referansları sağlayacaktır. Bir nokta (örneğin P1) müteakip deyimde referans verilmişse, hesaplamaya göre 2.75 değerindeki z koordinatı sağlanır.

ZSURF deyimini yalnızca nokta tanımını etkiler; fakat bir z koordinatını içeren bir POINT deyimini etkilemez.

4.11 APT GEOMETRİK TANIM DEYİMLERİ ÜZERİNE BAZI YORUMLAR

NC işlemcisi içinde yer alan bir APT geometrik tanım deyiminin işlevi, bir geometrik cisim hesaplamak ve tanımlamak için gerekli şartları sağlamaktır. Bu şartlar, kullanılan deyim formatları ile değişir. Ancak, her bir geometrik cisim tipinin tanımlanması için evvelki kısımlarda listesi verilmiş olan çeşitli formatlardan görülebileceği gibi, NC işlemcisinin belirli bir geometrik cisim tipini tanımlama yöntemi tektir. Bunun anlamı şudur: Verilen şartlara veya kullanılan deyim formatlarına bağlı olmaksızın; aynı tipteki geometrik öğeler, daima, kanonik biçim denilen standart bir eşitliğin katsayıları olan belirli bir parametre grubu tarafından tanımlanır (Ek B'ye bakınız). Verilen değişik şartlara dayalı olarak bu parametrelerin hesaplanması, NC işlemcisi tarafından yapılır. Örneğin, bir dairenin kanonik biçimi:

$$(x, y, z, a, b, c, r)$$

olup burada; x, y ve z merkez noktasının koordinatları; a, b ve c ise, daireyi içeren düzleme dik konumdaki birim vektörün X, Y ve Z koordinatlarıdır. NC işlemcisinde bir daire, Z eksenine paralel bir silindirdir olduğu için; a, b ve c sırasıyla 0, 0 ve 1'dir. Kanonik biçimdeki r değeri ise dairenin yarıçapıdır. Bir daire, çember üzerindeki verilen üç nokta tarafından tanımlanabilir. Bu durumda, merkezin koordinatları ve yarıçapın, verilen üç noktanın koordinatları vasıtasıyla hesaplanması gerekir. Bazen de bir daire, verilen üç teğet doğru ile tanımlanır. Bu duruma göre hesaplama, kanonik biçimdeki parametreleri elde etmek üzere, farklı bir yordamın kullanılmasıyla yapılmalıdır. Bu da açıkça göstermektedir ki; geometrik tanım deyimleri, gerekli matematiksel hesap işlemleri hakkında bilgi sahibi olmadan bile bir geometrik cisim tanımlayabilmemizi sağlayan birer güçlü araçtır.

Geometrik tanım deyimlerinin bir çoğu için, verilen şartları sağlayan sadece bir tek çözüm vardır. Bununla beraber, birden çok sayıda çözüm ortaya çıkaran şartlara bağlı olarak geometrik cisimleri tanımlayan çok sayıda deyim de mevcuttur. Bu durumda, mümkün olabilecek çözümler arasında isteneni seçmek için, deyim içinde ilave bir değiştirici ya da değiştiriciler kullanmak gerekir. Maalesef APT programlama kılavuzlarında, düzelticilerin kullanımı ile ilgili ayrıntılı açıklamalar yoktur.¹² Bu da, bazı durumlarda, doğrultucu değiştirici veya değiştiricilerin seçilmesinde karışıklığa yol açmaktadır. Aşağıda, bazı tipik örnekler verilmiştir.

Örnek 1

Tablo 4-7'deki No. 3 deyim formatı için, genellikle verilen durumu (çember üzerinde iki nokta ile yarıçap) sağlayan iki mümkün çözüm vardır. Bu duruma göre, bir doğrultu değiştiriciye gerek vardır ve bu deyim formatı, mümkün olan iki çözüm elde edilebilecek biçimde tasarlanmıştır. Verilen yarıçapın verilen iki nokta arasındaki uzaklığın yarısına eşit olduğu bir özel durumda ise, çözüm sayısı bire düşürülür. Bu duruma göre, doğrultu değiştiriciyi nasıl seçebiliriz?

Örnek 2

Tablo 4-10'daki No. 2 deyim formatı, verilen bir nokta, bir teğet çemberi ve bir yarıçap esasına bağlı olarak bir daireyi tanımlar. Bu verilen durumlar kümesi de şöyle bir karmaşık durum ortaya çıkarır: r ve d arasındaki ilişkiye bağlı olarak, çözüm sayısı sıfırdan dörde kadar değişebilir (Tablo 4-10'daki ilgili şekle bakınız). Bu durumda ortaya çıkan sorun şudur: Mümkün olan çözüm sayısı dörtten az olduğu takdirde, bu daire deyimindeki değiştiricileri nasıl belirtebiliriz?

APT programlama kılavuzunda anlamları net biçimde açıklanmamış olan birkaç geometrik tanım deyimini vardır. Bir silindirin merkezi olarak tanımlanan nokta (Tablo 4-2'deki No. 4 deyim formatı), bunun tipik bir örneğidir. Bir silindirin geometrik merkezi ne demektir? NC işlemcisinde, CENTER kelimesini içeren bir CYLNDR deyimiyile tanımlanan bir silindir, belirli bir merkeze sahiptir. Bununla beraber, içinde CENTER kelimesi yer almayan bir CYLNDR deyimiyile tanımlanan bir silindirin merkezi, APT programlama kılavuzlarında¹² herhangi bir inceleme veya açıklama mevcut değildir. Bölüm 4.2'de açıklandığı üzere, bir LINE deyimiyile tanımlanan geometrik öğenin, bazı durumlarda X - Y düzleminde bir doğru veya X - Y düzlemine dik bir düzlem olabilmesi de, diğer bir tipik örnektir.

Çok büyük sayıda geometrik cisim ve bunlara ait deyim formatı mevcut olması nedeniyle, bunların tümünü bu kitabın kapsamında listelemek ve açıklamak

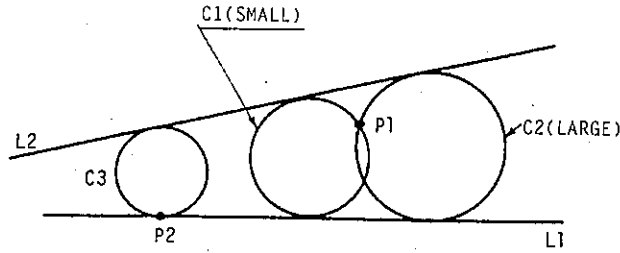
mümkün değildir. Ama yine de, aşağıda belirtilen yöntem ve kurallar, bir deyim detaylarını yazan bir programcıya yararlı olacaktır.

Bir değiştirici seçimi yaparken, programcının mümkün çözümlerin sayısı ve yerlerini açık biçimde anlamış olması kesinlikle şarttır. Bir kural olarak, herbir değiştirici mümkün olan iki durumu (örneğin; solda ve sağda, içinde ve dışında, artan X doğrultusu ve azalan X doğrultusu vb) birbirinden ayırdedilmesi gerekir. Mümkün iki durumun bire indirgenmesi halinde ise, değiştirici, belirli bir çözümün seçimi üzerinde hiçbir etkiye sahip değildir. Sonuç olarak, deyim formatında listelenen düzelticilerden herhangi biri seçilebilir. Örneğin bir dairenin; bir L1 teğet doğrusu, bir C1 teğet çemberi ve verilen bir r yarıçapı vasıtasıyla tanımlandığını varsayalım (Tablo 4-10'da No. 3 deyim formatına bakınız). L1 için YSMALL ve C1 için IN olmak üzere diğer iki değiştirici tarafından seçilmiş olan iki mümkün çözümden (C2 ve C3) istenileni seçmek için, bu deyimde L1'den hemen sonra gelen XLARGE,..., veya YSMALL doğrultu değiştirici kullanılır (Tablo 4-10'da No. 3 deyim formatına ait şekle bakınız). Eğer r yarıçapı $d/2$ 'ye eşitse, L1 doğrusunun alt tarafında ve C1 dairesinin içinde yer alan yalnızca bir teğet daire (bir tek çözüm) vardır. Yani, mümkün olan iki çözüm (C2 ve C3) bire düşürülmüş olur. Bu durumda, doğrultu değiştiricisinin, sonucun seçilmesi üzerinde hiçbir etkisi yoktur. Dolayısıyla, deyim içinde bu değiştiricilerin herhangi birisi belirtilebilir. Şekil 4-22'deki örnek de, diğer düzelticilerin belirtilmesinden sonra yalnızca bir tek çözüm kaldığı takdirde SMALL ve LARGE boyut değiştiricilerin hiçbir etkisinin olmayacağını göstermektedir.

Bir geometrik cismin doğru biçimde tanımlanmasını sağlayan yararlı bir yöntem de, tanımlanan C1, C2 vb. gibi geometrik cisimlerin kanonik biçimini yazmak için

PRINT/3,C1[,C2[,C3[,...]]]

deyimini kullanmaktır. Örneğin Tablo 4-10'daki No. 5 deyim formatında, L1 doğrusundan önce belirtilen XLARGE,..., ve YSMALL doğrultu değiştiricilerin işlevi, kullanım kılavuzunda iyi tanımlanmamış ve açık biçimde açıklanmamıştır. Bu değiştiricilerin, verilen bir L1 doğrusuna göre tanımlanan dairenin yerini göstermek için kullanıldıkları görülür. Bunların işlevini açık olarak anlayabilmek ve istenen daireyi doğru biçimde tanımlayabilmek için; yalnızca ihtiyaç duyulan geometrik tanım deyimlerini içeren, aşağıdaki gibi basit bir program yazabiliriz:



Şekil 4-22 İki teğet doğrusu (L1 ve L2) ve çember üzerinde verilen bir P1 noktası ile tanımlanan bir daire. Genellikle, C1 ve C2 olmak üzere iki çözüm vardır ve bu durumda, bir boyut değiştirici gereklidir. Eğer verilen nokta, doğruların biri üzerinde ise (örneğin P2 noktası); yalnızca bir çözüm vardır. Böyle bir durumda, LARGE ve SMALL boyut değiştiricilerin her ikisi de aynı sonucu (C3) verir.

.....

.....

L1 = LINE/X3,Y3,Z3,X4,Y4,Z4

C1 = CIRCLE/CENTER,X2,Y2,Z2,RADIUS,R1

P1 = POINT/X1,Y1,Z1

C2 = CIRCLE/LARGE,M1,TANTO,L1,IN,TANTO,C1,THRU,P1

PRINT/3,C2

.....

.....

Bu programdaki M1; XLARGE,..., veya YSMALL değiştiricisi olabilir. NC işlemcisi, programı işledikten sonra, program listeleme dosyasına kanonik biçimi yazacaktır. Bu programda yalnızca M1 değiştiriciyi değiştirmek suretiyle, dört kanonik biçim elde edebiliriz. Bu sonuçları karşılaştırarak, söz konusu değiştiricilerin işlevini belirleyebiliriz. Örneğin, eğer yukarıdaki programda:

X1 = 0; Y1 = 1; Z1 = 0;

X2 = 1; Y2 = 3; Z2 = 0;

X3 = Y3 = Z3 = 0;

X4 = Y4 = 1; Z4 = 0;

R1 = 1

ise, C2 dairesinin değişik değiştiricilerle meydana getirilen kanonik biçimleri aşağıdaki gibidir (yalnızca X, Y ve R parametreleri listelenmiştir):

M1	C2		
	X	Y	R
XLARGE	-32.3839	45.8525	55.3216
XSMALL	-32.3839	45.8525	55.3216
YLARGE	-32.3839	45.8525	55.3216
YSMALL	-32.3839	45.8525	55.3216

Test sonuçları açıkça göstermektedir ki; C2 dairesinin tanımlanmasında M1 doğrultu değiştiricisinin hiçbir etkisi yoktur. Aslında, doğrultu değiştirici, tanımlanan dairesinin diğer çözüme göre bağlı konumunu gösterir. Ancak bu iki çözümün aynı çapa sahip olması durumunda, değiştiricinin etkisi vardır (Problem 4.9'a bakınız). Burada açıklanan yöntem, bir silindirin merkezini belirlemek için de kullanılmış ve test sonuçlarına dayalı olarak, Tablo 4-2'deki 4 nolu deyim formatında belirtilen sonuca ulaşılmıştır.

Diğer bir geometrik cisim tanımlamak için kullanılan bir geometrik cisim, mutlaka önceden tanımlanmış olmalıdır. Aksi takdirde; tanımlanmamış olan bir geometrik cismin sembolünü yerine koymak üzere, bir deyimle içiçe bir geometrik tanımlanan deyim kullanılabilir. Örneğin, yukarıda listelenmiş olan programda, eğer

L1 = LINE/X3,Y3,Z3,X4,Y4,Z4

deyimi eksikse; C2 dairesini tanımlayan deyim aşağıdaki biçimde yazabiliriz:

C2 = CIRCLE/LARGE, XLARGE, TANTO, (LINE/X3, Y3, Z3, X4, Y4, Z4), \$
IN, TANTO, C1, THRU, P2

Mütekip deyimlerde doğruya atıfta bulunulmayacak ise:

C2 = CIRCLE/LARGE, XLARGE, TANTO, (LINE/X3, Y3, Z3, X4, Y4, Z4)\$
IN, TANTO, C1, THRU, P2

Mütekip deyimlerde L1 doğrusuna atıfta bulunulacak ise:

C2 = CIRCLE/LARGE, XLARGE, TANTO, (L1 = LINE/X3, Y3, Z3, X4, Y4, Z4)\$
IN, TANTO, C1, THRU, P2

Söz dizimi ve sayı hatalarının yanısıra, geometrik tanımlama deyimlerinde en sık karşılaşılan hatalar şunlardır:

1. Tanımlanan geometrik cisim irrasyoneldir. Örneğin bir nokta, birbirini kesmeyen bir doğru ile bir çemberin kesişimi olarak tanımlanmıştır.

2. Bir geometrik cisim, biri veya birkaçı tanımlanmamış olan diğer geometrik cisimlere dayanarak tanımlanmıştır.
3. Bir geometrik cisim yeniden tanımlanmıştır. NC işlemcisi, bir geometrik cismin sembolünün yeniden tanımlanabilmesine izin vermez. Örneğin, eğer P1 bir nokta (3,2,4) olarak tanımlanmış ise; diğer noktaları veya diğer geometrik cisim tiplerini ifade etmek üzere yeniden tanımlanamaz veya kullanılamaz. Eğer programlama gereği olarak bir sembolün yeniden tanımlanması zorunlu ise, sembolün yeniden tanımlanmasından önce

CANON/ON

deyiminin belirtilmesi gerekir. Bu deyim, NC işlemcisinin işlem modunu değiştirir. Sonuç olarak, geometrik cisimlere verilen semboller, kendisinden sonra gelen deyimlerde yeniden tanımlanabilir.

CANON/OFF

deyimi ise, CANON/ON deyimine zıt bir etkiye sahiptir. Bu deyim belirtildiği takdirde, NC işlemcisini, bir geometrik cisim sembolünün yeniden tanımlanabilme imkanı bulunmayan ön değer moduna döndürecektir.

4.12 MATEMATİKSEL HESAPLAMA FONKSİYONLARI

Bu kısımda, APT-AC NC işlemci kapsamına dahil edilmiş olan matematiksel hesaplama fonksiyonları tanıtılmaktadır. Evvelki, kısımlarda yapılan açıklamalardan anlaşılacağı üzere, bir geometrik cisim, bir veya birkaç geometrik parametre vasıtasıyla tanımlanır. Birçok durumda, bu parametrelerin elle hesaplanması karmaşık ve zaman alıcıdır. Dolayısıyla, NC işlemciyi yapısında bulunan matematiksel hesaplama fonksiyonları kullanılabilir.

Tablo 4-23, NC işlemcisi tarafından gerçekleştirilebilen matematiksel fonksiyonların bir listesi, herbir deyim için eşdeğer matematiksel ifade ile birlikte vermektedir. Bir matematiksel fonksiyon tarafından işletilmesi gereken bir argüman, fonksiyonu tanımlayan sözcükten sonraki bir parantez çifti içine dahil edilmelidir. Argüman; bir sayı olabileceği gibi, bir değişkeni temsil eden bir sembol, veya bir matematiksel ifade olabilir. Bazı argüman örnekleri şöyle olabilir:

SINF(30)

SINF(A) (A, değeri bir önceki deyim vasıtasıyla verilmiş olan bir değişkendir.)

SINF(A-B) (A ve B değerleri, önceki deyimler vasıtasıyla verilir.)

Deyimlerin bazıları, argüman olarak sadece geometrik cisimlerin sembollerini kabul edebilir. Bu deyimler; DOTF, LNTHF, ANGLF ve DISTF'dir.

Geometrik cisimlerin tanımlanmasında matematiksel hesaplama deyimlerinin kullanımına ait örnekler, Bölüm 4.13'de verilmiştir.

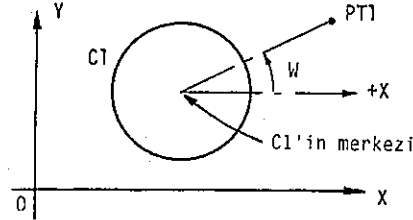
Matematiksel fonksiyonların kullanımı, geometrik cisimlerin tanımlanması ile sınırlı değildir. Bu fonksiyonlar, programlamada ihtiyaç duyulan herhangi bir hesaplama için de kullanılabilir.

TABLO 4-23 MATEMATİKSEL İŞLEMİ TANIMLAYAN DEYİMLER

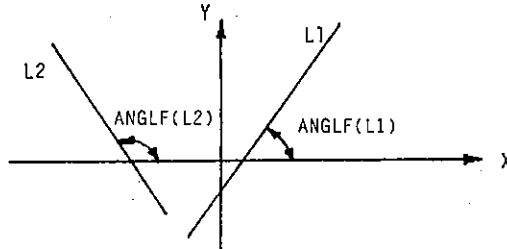
Deyim	Eşdeğer Matematiksel İfade ve Anlamı
Trigonometrik fonksiyonlar	
SINF (A)	sinA
COSF (A)	cosA
TANF (A)	tanA
COTANF (A)	cotA
ASINF (A)	arcsinA $(-90^{\circ} \leq \text{ASINF}(A) \leq 90^{\circ})$
ASINF (A)	arccosA $(0^{\circ} \leq \text{ACOSF}(A) \leq 180^{\circ})$
ACOSF (A)	arctanA $-90^{\circ} \leq \text{ASINF}(A) \leq 90^{\circ}$
ATANF (A)	arctan (A/B) $-90^{\circ} \leq \text{ASINF}2F(A,B) \leq 90^{\circ}$
ATAN2F (A,B)	
Diğer fonksiyonlar	
ABSTF (A)	A'nın mutlak değeri
CBRTF (A)	$A^{1/3}$
EXPF (A)	e^A
INTGF (A)	A giriş argümanı, bir tam sayıya çevresel sınırlandırılmıştır.
LOGF (A)	lnA
LOG10F (A)	$\log_{10} A$
MAXIF (A1,A2,...,An)	A1, A2,...,An'nin en büyüğü
MINIF (A1,A2,...,An)	A1,A2,...,An'nin en küçüğü
MODF (A1,A2)	A1/A2 bölümündeki kalan
SIGNF (A1,A2)	Birinci argümanın mutlak değerine ikinci argümanın işaretini verme; yani (A2'nin işareti). $\emptyset A1\emptyset$.
SQRTF (A)	$A^{1/2}$
DOTF (V1,V2)	Tanımlanmış iki vektörün (V1 ve V2) skalar (yönsüz) çarpımı. Eğer $V1 = a_1.i + b_1.j + c_1.k$ ve $V2 = a_2.i + b_2.j + c_2.k$ ise; $\text{DOTF}(V1,V2)$, $a_1a_2 + b_1b_2 + c_1c_2$ 'ye eşdeğerdir.
LNTHF (V1)	V1 vektörünün büyüklüğü: $(a_1 + b_1 + c_1)^{1/2}$
ANGLF	Birkaç geometrik cisim arasındaki açısal ilişki, bu fonksiyon vasıtasıyla elde edilebilir.

TABLO 4-23 MATEMATİKSEL İŞLEMİ TANIMLAYAN DEYİMLER (Devam)

Deyim	Eşdeğer Matematiksel İfade ve Anlamı
1. ANGLF(C1,PT1)	C1: tanımlanmış bir daire, PT1: tanımlanmış bir nokta olup; bu deyim tarafından tanımlanan açı W'dir.



2. ANGLF(PT1,PT2,PT3) PT1,PT2,PT3: tanımlanmış noktalardır. Bu deyim, PT1PT2 ve PT2PT3 doğruları arasında kalan daha küçük açıyı (mutlak değer olarak) tanımlar.
3. ANGLF(L1) L1: tanımlanmış bir doğrudur. Bu deyim; L1 doğrusunun, XY işleminin üst kısmındaki pozitif X eksenine ile yaptığı açıyı (mutlak değer olarak) tanımlar.



4. ANGLF(PL1,PL2) Tanımlanmış iki düzlem (PL1 ve PL2) arasında kalan daha küçük açıyı (mutlak değer olarak) tanımlar.
5. ANGLF(L1,L2) Tanımlanmış iki doğru (L1 ve L2) arasında kalan daha küçük açıyı (mutlak değer olarak) tanımlar.
6. ANGLF(d,m,s) Bir açıyı derece (d), dakika (m) ve saniye (s) cinsinden tanımlar.
- DISTF(G1,G2) Paralel iki doğru, iki düzlem veya bir doğru ve bir düzlem arasındaki uzaklık (mutlak değer).
- G1 ve G2: Tanımlanmış doğrulara veya düzlemlere ait sembollerdir. DISTF(P1,G1) Bir P1 noktası ile; bir nokta, doğru, düzlem, daire, silindir veya küre olabilen bir G1 geometrik cismi arasındaki uzaklık (mutlak değer).
- TYPEF(A) A: Bir sayı, kelime veya anlamlı bir alfasayısal dizi.

Eğer	$A \leq 1.0 \cdot 10^{-14}$ ise	TYPEF(A) = 1
	$A > 1.0 \cdot 10^{-14}$ ise	TYPEF(A) = 1
	A bir kelime ise	TYPEF(A) = 3
	A alfasayısal ise	TYPEF(A) = 4

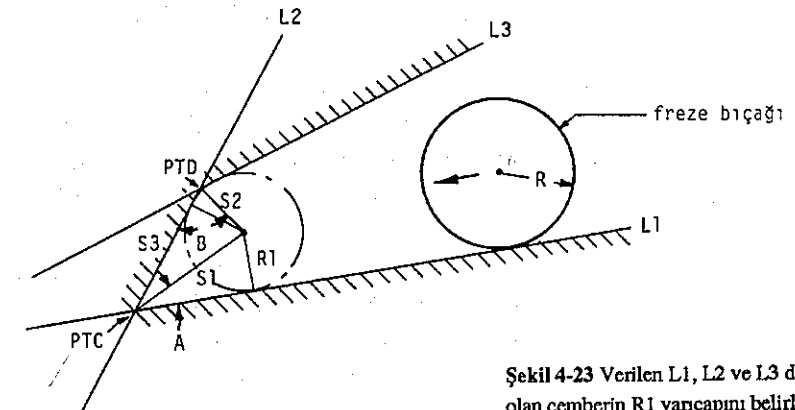
TABLO 4-23 MATEMATİKSEL İŞLEMİ TANIMLAYAN DEYİMLER (Devam)

Deyim	Eşdeğer Matematiksel İfade ve Anlamı
İlişkisel İşleç	
'EQ'	Eşittir
'GE'	Büyük veya eşittir
'LE'	Küçük veya eşittir
'NE'	Eşit değildir
'GT'	Büyüktür
'LT'	Küçüktür
Örnek	C = A'EQ'B deyimi; eğer A = B doğru ise C = 1, eğer A = B doğru değilse C = 0 olduğunu ifade eder.

Şekil 4-23, bir örneği göstermekte olup; parça profilinin kesiti, L1, L2 ve L3 olmak üzere, üç doğrudan oluşmaktadır. Freze kesicisi hareket güzergahını doğru biçimde tanımlayabilmek için, R yarıçaplı kesicisinin L1 doğrusu boyunca sola hareketi esnasında L2 ve L3 doğrularından hangisine ilk olarak temas edeceğini bilmek şarttır. Üç değişik durum için matematiksel şart şöyledir:

1. Eğer $R < R1$ ise, kesici ilk olarak L2 doğrusuna temas edecektir.
2. Eğer $R = R1$ ise, kesici ilk olarak hem L2 ve hem de L3 doğrusuna temas edecektir.
3. Eğer $R > R1$ ise, bıçak ilk olarak L3 doğrusuna temas edecektir.

Burada R1; L1, L2 ve L3 doğrularının her üçüne de teğet olan ortak çemberin yarıçapıdır. Dolayısıyla, bıçak güzergahının tanımlanması ile ilgili problemlerden biri; tanımlanmış üç doğru (L1, L2 ve L3) esasına dayalı olarak R1 yarıçapını



Şekil 4-23 Verilen L1, L2 ve L3 doğrularına teğet olan çemberin R1 yarıçapını belirlemek için, matematiksel deyimlerin kullanılması.

hesaplamaktır.

Şekil 4-23'den;

$$R1 = S1 \cdot \sin A = S2 \cdot \sin B$$

ve

$$S3 = S1 \cdot \cos A + S2 \cdot \cos B \text{ dir.}$$

olup burada;

S1 = PTC noktası ile ortak teğet çemberin merkezi arasındaki uzaklık
S2 = PTD noktası ile ortak teğet çemberinin merkezi arasındaki uzaklık
S3 = PTC ve PTD noktaları arasındaki uzaklık

Yukarıda sıra ile belirtilen eşitliklerin çözümü:

$$R1 = S3 / (\cot A + \cot B)$$

Dolayısıyla, APT programının, tanımlanmış L1, L2 ve L3 doğrularına bağlı olarak R1 yarıçapını belirleyen kısmı, aşağıdaki gibi olabilir:

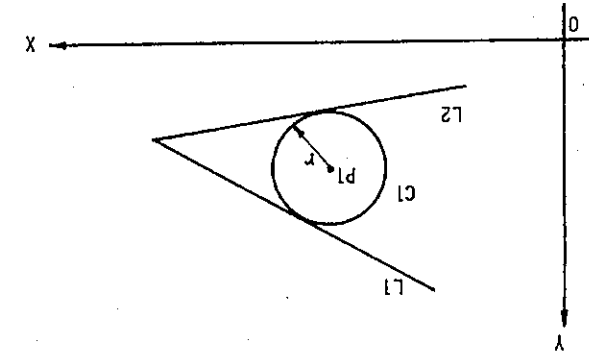
```
.....
.....
..... $$ BU KISIMDA L1, L2 ve L3 DOĞRULARI TANIMLANIR
A = (ANGLF (L1,L2))/2
B = (180 - ANGLF (L2,L3))/2
PTC = POINT/INTOF,L1,L2
PTD = POINT/INTOF,L2,L3
V1 = VECTOR/PTC,PTD
S3 = LNTHF (V1)
R1 = S3/(COTANF (A)+COTANF (B))
.....
.....
```

4.13 GEOMETRİK TANIMLAMA DEYİMLERİNİN UYGULAMALARI

4-1'den 4-22'ye kadar (dahil) olan tablolar, APT'de izin verilen geometrik öğeleri tanımlamak için ihtiyaç duyulan deyim ve formatların çoğunu listelemektedir. Bu çeşitli deyim formatları, bir programcının geometrik cisimleri tanımlayabilmesi için önemli bir serbestlik derecesi sağlar. Bir geometrik cisim; önceden bildiğimiz bilgilere dayalı olarak veya dahası, önceki deyimlerde tanımlanmış olan geometrik cisimleri esas almak suretiyle tanımlanabilir. Bu ikincisinin kullanımı, gerekli hesaplamaların sayısını büyük ölçüde azaltabilir. Örneğin, bir parmak frezeyi, iki doğru çizgi tarafından oluşturulan bir köşenin en iç konumuna kadar (Şekil 4-24) hareket ettirmek zorunda olduğumuzu, fakat bu noktanın tam konumunu bilmediğimizi varsayalım. APT programında, kesici takımın bu noktaya gitmesi için gerekli konum ve işlem sırasını, hiçbir hesap işlemine gerek kalmadan

belirleyebiliriz. İşlem sırası aşağıdaki gibidir:

1. LINE deyimini kullanarak, verilen şart esasına dayalı olarak L1 ve L2 doğrularını tanımlayınız.
2. Yarıçapı parmak freze yarıçapına eşit olmak üzere, bu iki doğruya teğet olan ve herbir doğrunun istenen tarafında bulunan C1 dairesini tanımlayınız.
3. C1 dairesinin merkezi olarak P1 noktasını tanımlayınız
4. Kesici takımın P1 noktasına gitmesi amacıyla komut vermek için, bir noktadan noktaya hareket deyimini (GOTO)* yazınız.



Şekil 4-24 İki doğruya (L1 ve L2) teğet olan bir daire

Buna ait program kısmı şöyle görünebilir:

```
.....
.....
L1=LINE/.....
L2=LINE/.....
C1=CIRCLE/YSMALL,L1,YLARGE,L2,RADIUS,r
P1=POINT/CENTER,C1
.....
.....
GOTO/P1
.....
.....
```

Bölüm 5'te açıklanan hareket başlatma deyimini kullanılarak, bu hareket daha da kolaylıkla gerçekleştirilebilir.

Formatı:

* Hareket deyimleri Bölüm 5'te incelenmektedir.

OBTAIN, $\left\{ \begin{array}{l} \text{POINT} \\ \text{LINE} \\ \text{CIRCLE} \\ \dots \\ \dots \end{array} \right\} / C1, [A1], [A2], \dots, An$

olan OBTAIN deyimini kullanmak suretiyle de, bir geometrik cismin kanonik biçim parametrelerini elde edebiliriz. Bu deyimdeki ikinci eleman, C1 sembolü ile gösterilen bir geometrik cisim tipini belirten kelimedir. Kanonik parametrelerin elde edileceği C1 geometrik cisminin bir önceki deyimde tanımlanmış olması gerekir. A1, ..., An elemanları; C1 geometrik cisminin sırasıyla birinci, ikinci, ..., n'inci son kanonik parametrelerine atanacak sembollerdir. Eğer kanonik parametrelerin sadece bazılarında ihtiyaç varsa, yalnızca o parametrelere ait semboller belirtilir (virgüller ve son sembol -An- atlamaz). Böylece; önceki örnekteki daire merkez noktasının (P1) x ve y koordinatları, X1 ve Y1'i P1 merkez noktasının x ve y koordinatlarına ait semboller (skalar değişken) olarak atayan

OBTAIN,CIRCLE/C1,X1,Y1.....R1

deyiminin kullanılmasıyla elde edilebilir. Bundan sonraki deyimlerde de, basit olarak X1 veya Y1'i belirtmek suretiyle P1 noktasının x veya y koordinatına atıfta bulunabiliriz.

Bir geometrik cismin tanımlanması için ihtiyaç olunan verileri hesaplamak için de, APT içinde sağlanmış olan matematiksel hesaplama fonksiyonlarından yararlanabiliriz. Şekil 4-25'te APT matematiksel hesaplama deyimleri kullanılarak yapılan bir geometrik cisim tanımlaması ile ilgili bir örnek verilmektedir. İş parçası,

$$x = (3/2) \cdot R \cdot \cos(a) + (R/4) \cdot \cos(3a)$$

$$y = (3/2) \cdot R \cdot \sin(a) + (R/4) \cdot \sin(3a)$$

parametrik eşitliği ile tanımlanabilen, epitrokoid (dış teker eğrisi) profilini bir Wankel (rotatif) motor hücresidir. Bu eşitlikte, R sabit çemberin yarıçapıdır. Eğrinin eğimi

$$dy/dx = -[\cos(a) + (1/2) \cos(3a)] / [\sin(a) + (1/2) \sin(3a)]$$

ifadesiyle tanımlanır.

APT'de bir epitrokoidin tanımlanması için hiçbir deyim bulunmadığından, P1'den P36'ya kadar (dahil) olan noktaların x ve y koordinatlarını hesaplarız. Sonra da LCONIC deyimini kullanarak profili tanımlarız. Bir noktadan diğerine artışı parametrik açının 10 derece ve R'nin de 40 mm olduğunu varsayalım. Program şöyle olabilir:

.....
 X1=60*COSF (10) + 10*COSF (30)
 Y1=60*SINF (10) + 10*SINF (30)
 X2=60*COSF (20) + 10*COSF (60)
 Y2=60*SINF (20) + 10*COSF (60)

.....
 X36=60*COSF (360) + 10*COSF (1080)
 Y36=60*SINF (360) + 10*SINF (1080)
 P1=POINT/X1,Y1
 P2=POINT/X2,Y2

.....
 P36=POINT/X36,Y36
 E1=LCONIC/5PT,P1,P2,P3,P4,P5
 E2=LCONIC/5PT,P5,P6,P7,P8,P9

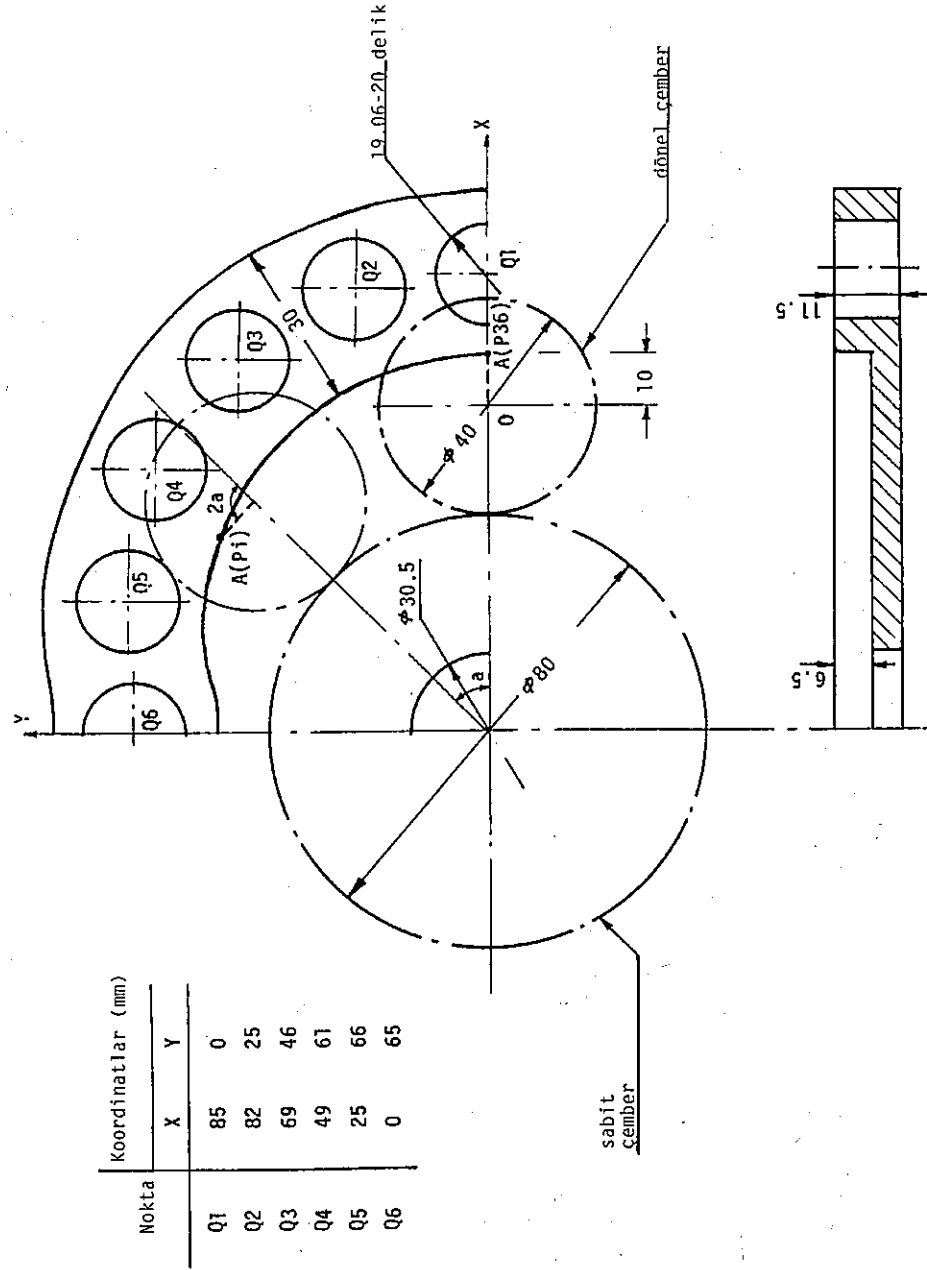
.....
 E9=LCONIC/5PT,P33,P34,P35,P36,P1

Bu programda, rotatif motor hücresinin profili dokuz kısma (yani E1'den E9'a kadar olan loft koniklere) bölünmüş olup; herbir bölüm, bir LCONIC/5PT deyimleriyle tanımlanmaktadır. İki bölümün birleşme noktasında eğim sürekliliğini garanti edebilmek için, bu deyimlerin yerine, LCONIC/3PT2SL deyimlerinin kullanılması gerekir. Wankel motor profilinin tanımlanmasına ait ayrıntılı açıklama, Bölüm 8.4'te verilmiştir.

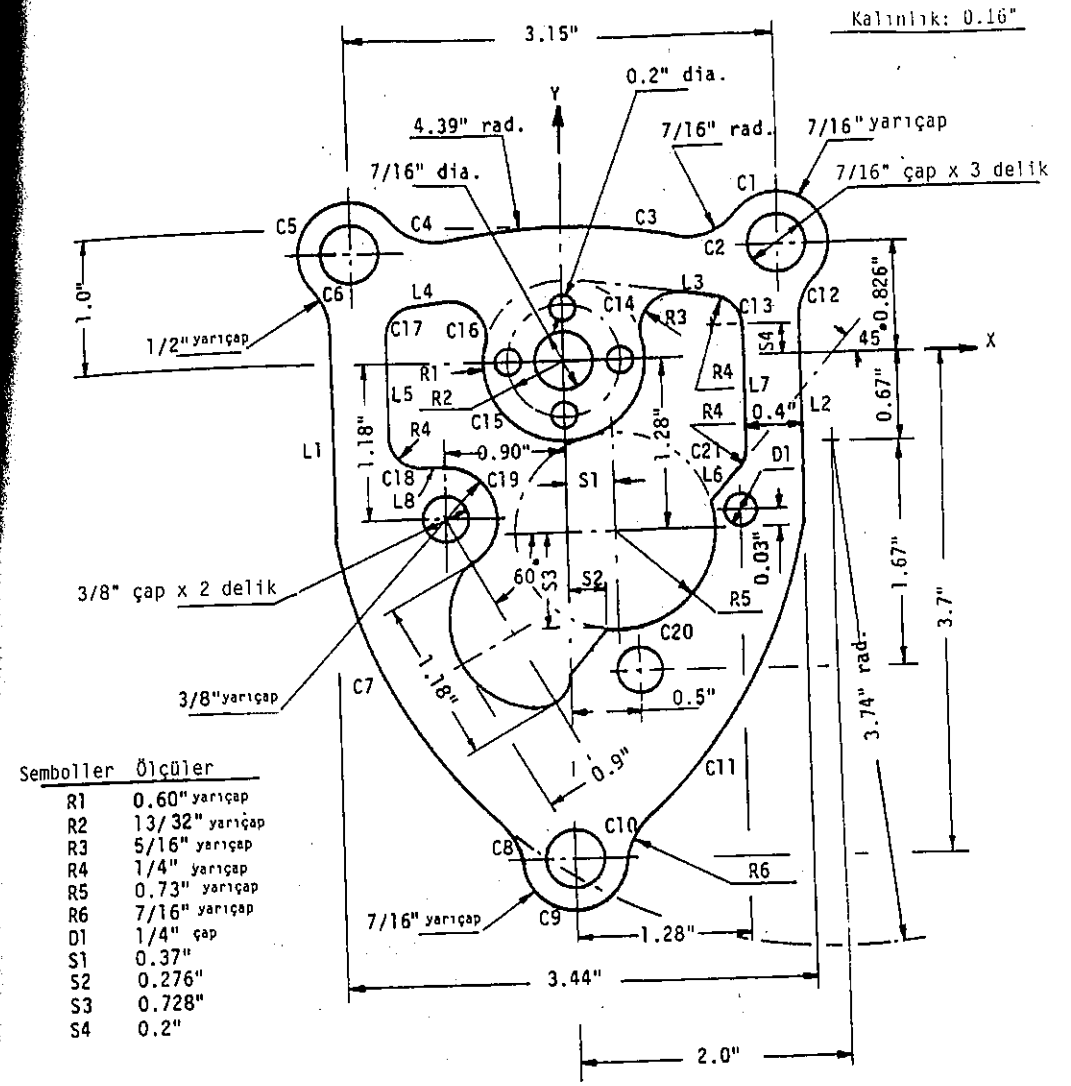
Epitrokoidal profili, TABCYL geometrik tanımlama deyimini vasıtasıyla da tanımlamak mümkündür.

Bu bölümü tamamlarken, bir geometrik cisim tanımlanması ile ilgili işlemleri açıklamak için, Şekil 4-26'da verilen iş parçasını kullanacağız. Parçanın iç ve dış profilleri, bir freze tezgahında işlenerek verilecektir. Dış profil, doğrulardan ve farklı yarıçaplardaki daire yaylarından oluşmakta; iç profil ise, daire yayları, elips yayları ve doğrulardan meydana gelmektedir. İşleme süresince parmak frezeyi, bu geometrik öğeler yönlendirmektedir.

Öncelikle programcının, parçanın geometrik şekillerini ve elemanlarını tanımlamak üzere en uygun koordinat sistemini seçmesi şarttır. Buna göre, şekilde gösterilen ve başlangıç noktası parçanın üst yüzeyinde bulunan X-Y koordinat sistemi seçilir. Sonra da, uygun geometrik tanımlama deyimleri ve matematiksel hesaplama deyimleri kullanılarak profillerin elemanları tanımlanabilir. Tanımlanmamış bir elemanı yerine koymak ve sonra da tanımla basitleştirmek için iç içe ifade de kullanılabilir. APT programının, freze bıçağı hareket güzergahının belirtilmesi için ihtiyaç olunan elemanları tanımlayan kısmı aşağıdaki gibi olabilir:



Şekil 4-25 Bir Wankel (rotatif) motor hüeresi. Hüerenin iç profili, dönel çember üzerinde hareket eden bir A noktasıyla meydana getirilen bir epitrokoidtir.



Şekil 4-26 Doğrular, daire yayları ve bir eliptik yaydan oluşan profilde bir taban plakası.

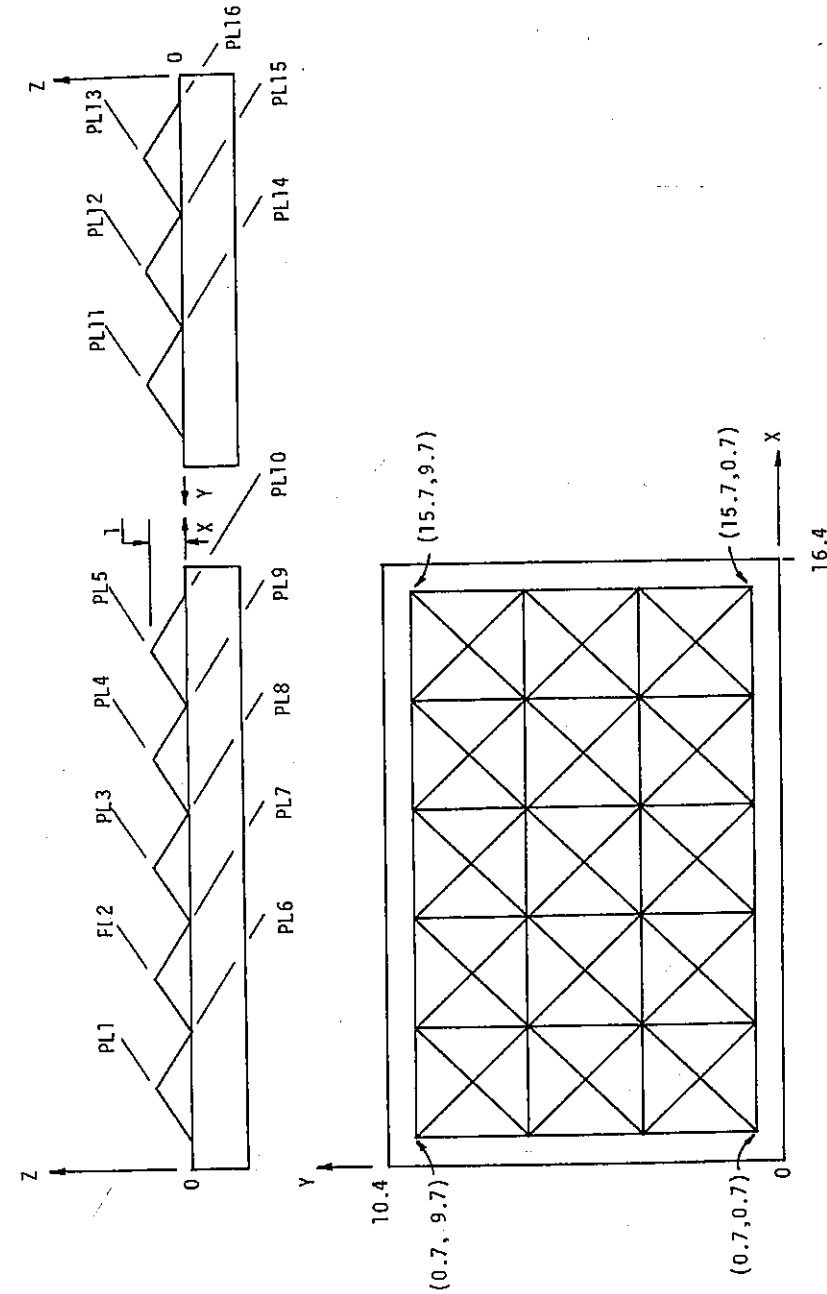
.....

 C1=CIRCLE/ (3.15/2),0.826, (7/16)
 C3=CIRCLE/0,-3.39,4.39
 C5=CIRCLE/(-3.15/2),0.826, (7/16)
 C2=CIRCLE/YLARGE, OUT, C1 OUT, C3,RADIUS, (7/16)
 C4=CIRCLE/YLARGE, OUT, C3, OUT, C5 RADIUS, (7/16)
 L1=LINE/YAXIS,-1.72
 L2=LINE/YAXIS,1.72
 C6=CIRCLE/XSMALL, L1,YSMALL,OUT,C5,RADIUS,0.5

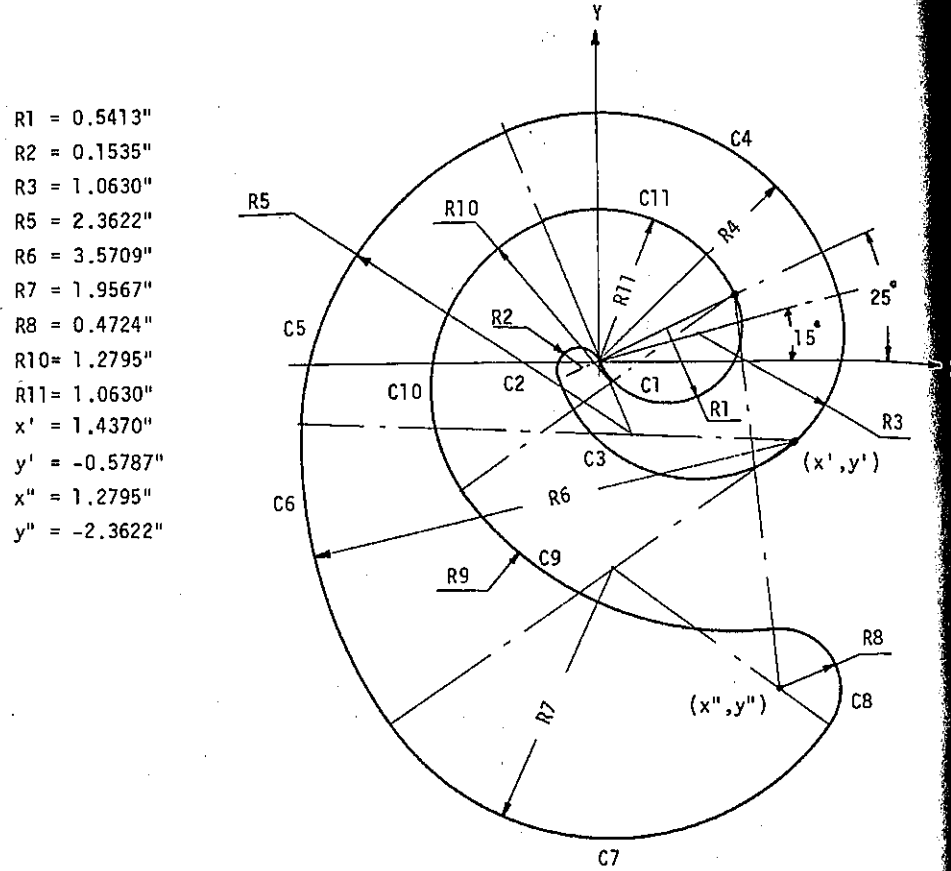
C12=CIRCLE/XLARGE,L2,YSMALL,OUT,C1,RADIUS,0.5
 C7=CIRCLE/2.0,-0.67,3.74
 C11=CIRCLE/-2.0,-0.67,3.74
 C9=CIRCLE/0.-3.7,(7/16)
 C8=CIRCLE/XSMALL,OUT,C9,OUT,C7,RADIUS,(7/16)
 C10=CIRCLE/XLARGE,OUT,C9,OUT,C11,RADIUS,(7/16)
 C13=CIRCLE/(1.72-0.4-0.25),0.2,0.25
 C17=CIRCLE/-(1.72-0.4-0.25),0.2,0.25
 C15=CIRCLE/0,0,0.6
 L3=LINE/LEFT,TANTO,C15,LEFT,TANTO,C13
 L4=LINE/RIGHT,TANTO,C15,RIGHT,TANTO,C17
 C14=CIRCLE/YSMALL,L3,XLARGE,OUT,C15,RADIUS,(5/16)
 C16=CIRCLE/YSMALL,L4,XSMALL,OUT,C15,RADIUS,(5/16)
 L7=LINE/YAXIS,1.32
 L5=LINE/YAXIS,-1.32
 X1=-0.9+(3/8+0.59)*COSF(60)
 Y1=-1.18-(3/8+0.59)*SINF(60)
 E1=ELIPS/CENTER,(POINT/X1,Y1),0.59,0.45,-60
 C19=CIRCLE/-0.9,-1.18,(3/8)
 L8=LINE/YLARGE,TANTO,C19,ATANGL,0
 C18=CIRCLE/YLARGE,L8,XLARGE,L5,RADIUS,0.25
 C20=CIRCLE/0.37,-1.28,0.73
 L6=LINE/0.276,-(1.28+0.728),ATANGL,45,XAXIS
 C21=CIRCLE/XSMALL,L7,YLARGE,L6,RADIUS,0.25

PROBLEMLER

- 4.1 Şekil P4-1'de gösterilen parça, oluklu (on düleli) bir yüzeye sahiptir. Bu yüzeyi oluşturan geometrik cisimleri tanımlayınız.
- 4.2 Şekil P4-2'de gösterilen şablonun profili, bir dizi daire yaylarından oluşmaktadır. Bu dairesel elemanları tanımlayınız.
- 4.3 Bir makaralı zincir sistemindeki zincir dişlisinin diş biçimi Şekil P4-3'te gösterilmiştir. ANSI Standard B29.1-1975'e göre, her dişin her iki yan yüzey profili de üç tane daire yayı ve bir tane doğrudan ibarettir (örneğin C3,C4,C5 ve L2). Aşağıda verilen şartlara göre C1-L1-C2-C3-C4-L2-C5 diş profilini tanımlayınız:



Şekil P4-1



Şekil P4-2

$$A = 39^\circ$$

$$B = 14.27^\circ$$

$$\text{Diş sayısı } N = 15$$

$$\text{Adım } P = 1/2 \text{ inç}$$

$$\text{Nominal makara çapı } D_r = 0.306 \text{ inç}$$

$$R_1 = (1.005 \cdot D_r + 0.003)/2 \text{ inç}$$

$$R_2 = D_r[0.8 \cdot \cos(18^\circ - 56^\circ/N) + 1.4 \cdot \cos(17^\circ - 64^\circ/N) - 1.3025] - 0.0015 \text{ inç}$$

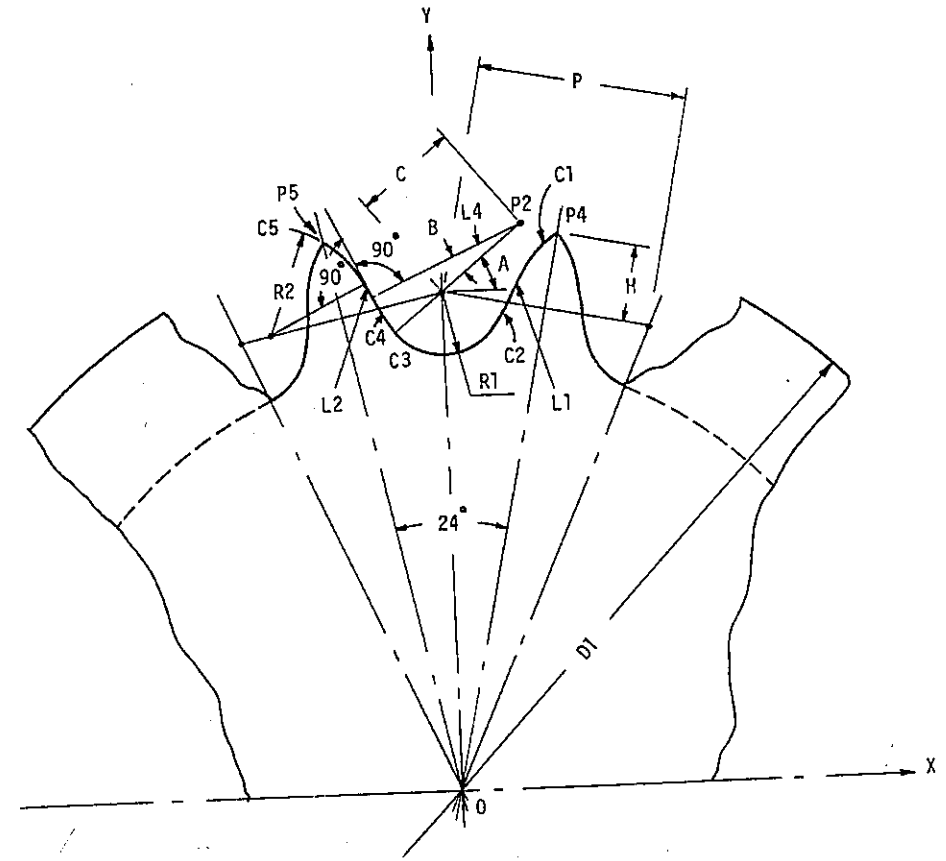
$$H = [(R_2)^2 - (1.4 \cdot D_r - 0.5 \cdot P)^2]^{1/2}$$

$$D_1 = P \cdot \cot(180^\circ/N) + (2 \cdot R_1 - D_r) \cos(180^\circ/N) + 2 \cdot H$$

$$C = 0.8 \cdot D_r$$

4.4

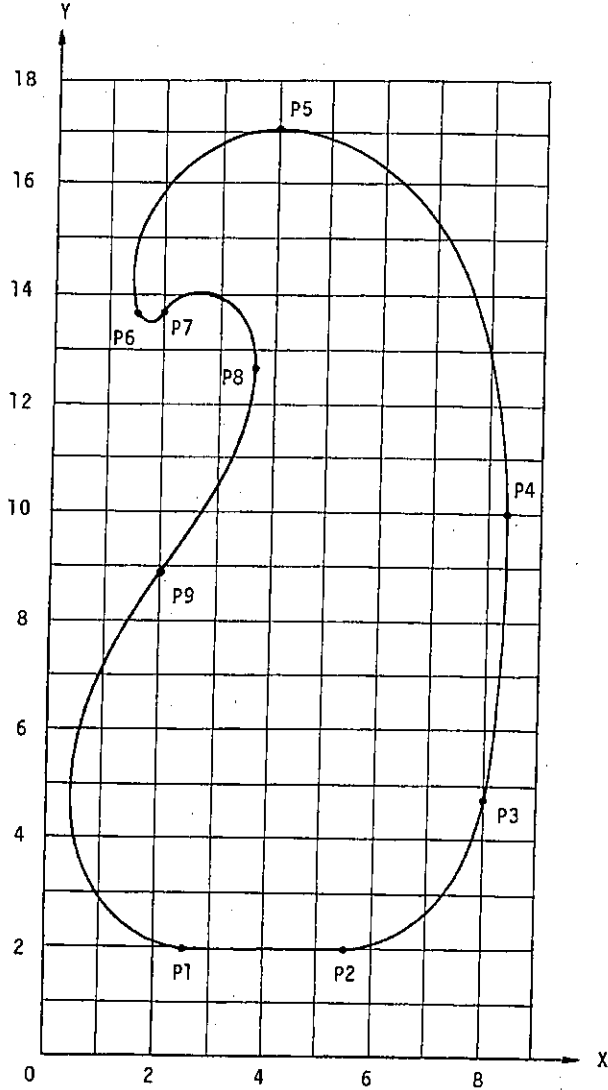
Şekil P4-4'de gösterilen ve bir sanatkarın elinden çıkmış olan bir proje resmine göre, parçanın bir SD (sayısal denetimli) freze tezgahında işlenerek yapılması istenmektedir. Profilin, P1'den P9'a kadar (dahil) olan noktalar vasıtasıyla dokuz kısma bölünmesi gerektiğini; ve P1-P2 bölümünün bir doğru, P6-P7 bölümünün ise bir çemberle yaklaşık duruma getirilebileceğini varsayınız. Geriye kalan bölümler LCONIC veya TABCYL deyimleriyle yaklaşık biçimde tanımlanabilir. Bir bölümü tanımlamak için, gerekiyorsa, söz konusu bölüm içindeki noktaların koordinatlarını görsel olarak tahmin ediniz. P4 ve P8 noktalarında profile teğet olan doğrular Y eksenine paraleldir. Profil üzerindeki P9 bükülme noktası, komşu bölümleri tanımlamak için LCONIC deyimini kullanıldığı



Şekil P4-3

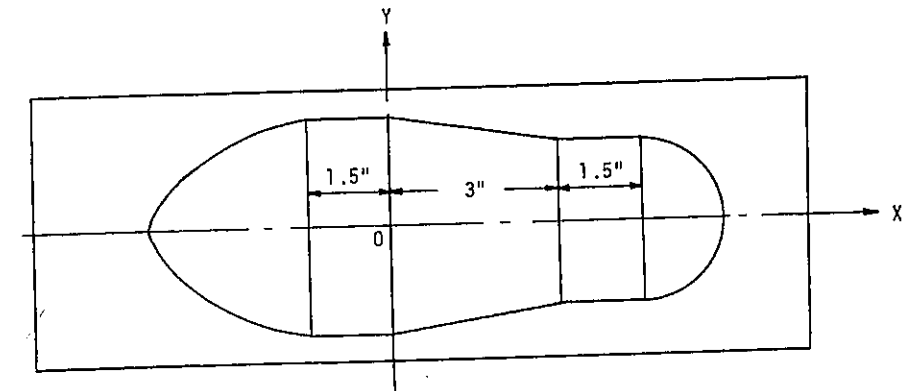
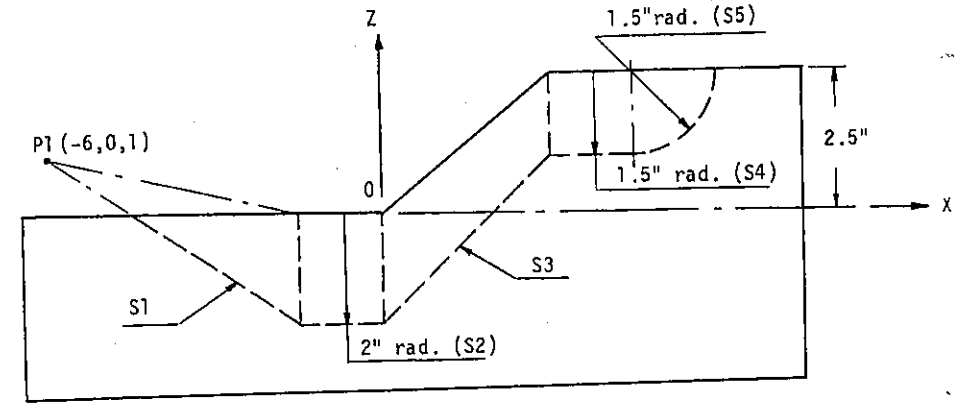
takdirde irrasyonel bir çözümden kaçınabilmek üzere, bölme noktası olarak seçilir. Profilin $z = 0$ düzleminde olduğunu da varsayınız. Birinden diğerine yumuşak bir geçiş sağlayacak biçimde, bu profilin dokuz bölümünü tanımlamak için uygun APT deyimlerini kullanınız.

- 4.5 Bir matris boşluğu (Şekil P4-5); taşınmış iki yüzey (S1 ve S3), iki dairesel silindir (S2 ve S4) ve bir küreden (S5) oluşmaktadır. Bu boşluğun profilini tanımlayınız.



Şekil P4-4

- 4.6 Bir parçanın profili (Şekil P4-6); bir elips (E1), merkezi (7.5,7.5) noktasında ve yarı kesit ve yarı eşlenik eksenleri sırasıyla 13 ve 4.5'e eşit olan iki yapraklı bir hiperbol (H1) ve LC1,LC2,LC3,LC4 geçiş eğrilerinden ibarettir. Bu geçiş eğrilerinin bitişik eğrilere teğet konumda olmalarını sağlamak için, LCONIC/3PT2SL deyimleriyle tanımlanması istenmektedir. Teğet noktaları, P1'den P8'e kadar (dahil) olan noktalardır. Uygun APT deyimlerini kullanarak bu profili tanımlayınız.
- 4.7 Bir çizelgelenmiş silindir Z eksenine paralel olup, Şekil P4-7'de verilen noktalara göre tanımlanmaktadır. Ayrıca; silindirin, C1 çemberine P1 noktasında ve L1 doğrusuna da P9 noktasında teğet olması gerekmektedir. Bu çizelgelenmiş silindiri tanımlayınız ve basılı çıktıyı kontrol edip doğrulayınız.
- 4.8 Şekil P4-8'de gösterilen bir matrisin boşluğu, S1 ve S2 olmak üzere iki parçaya bölünebilmektedir. POLCON deyimini kullanarak, bu iki parçanın yüzeylerini tanımlayınız. Aşağıdaki şartlar verilmiştir:



Şekil P4-5

$$C3_z = \begin{cases} -1 \\ -1 + 0.6t \end{cases}$$

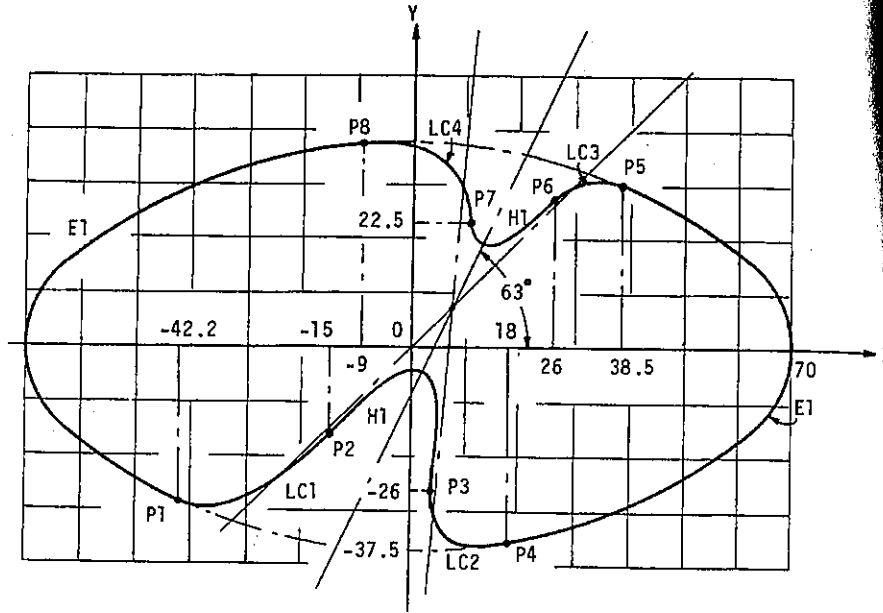
(S1 yüzeyi için)

(S2 yüzeyi için)

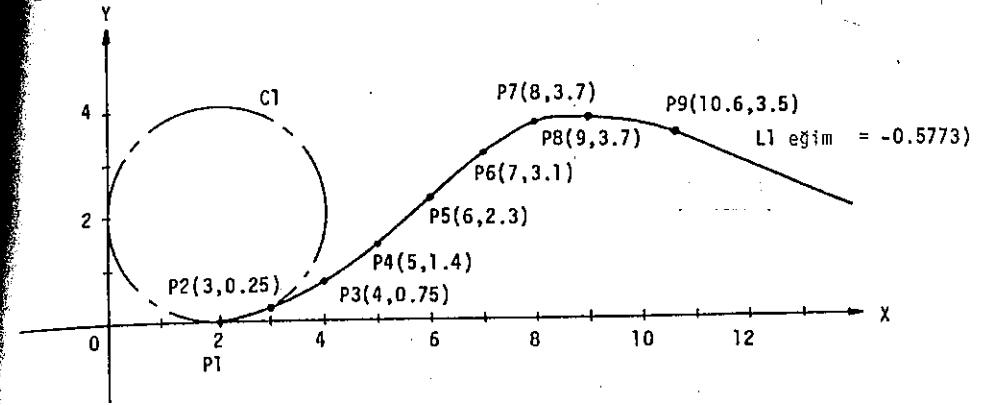
$$K = \begin{cases} 0.6668 \\ 0.6668 - 0.41t \end{cases}$$

(S1 yüzeyi için)

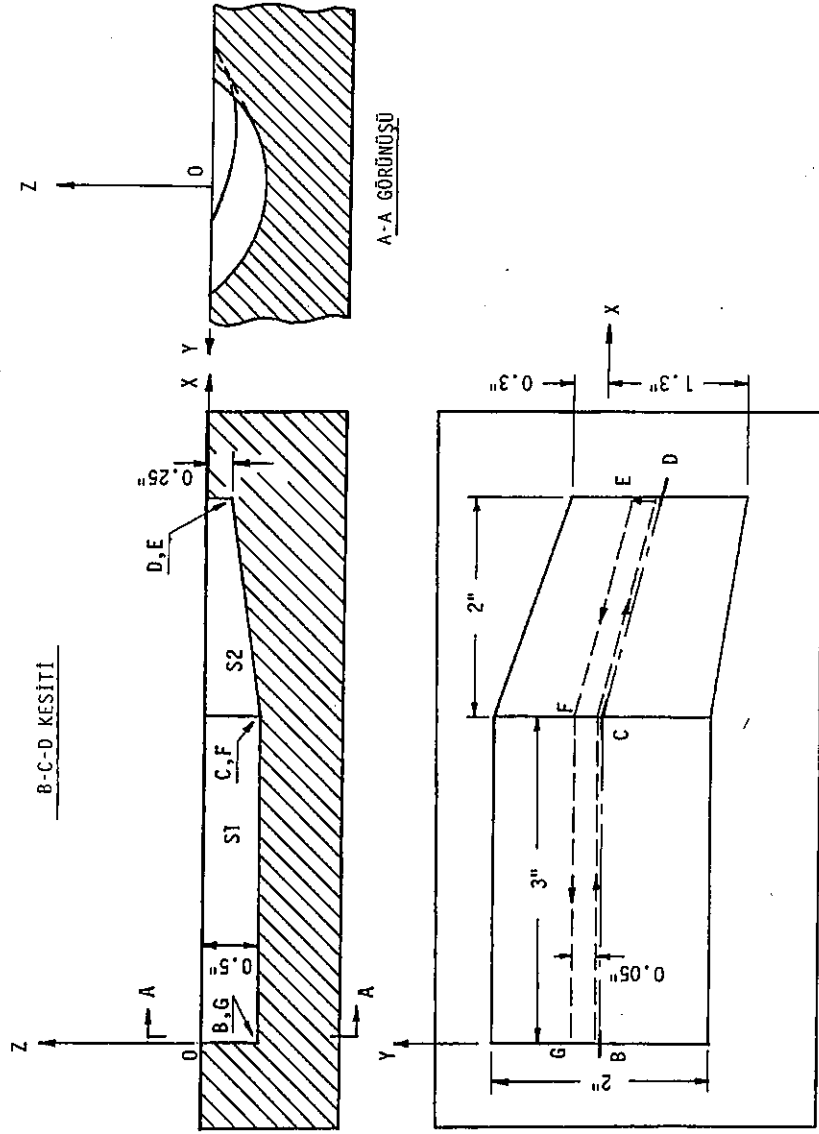
(S2 yüzeyi için)



Şekil P4-6



Şekil P4-7



Şekil P4-8

Bölüm 5

Kesici Hareketinin Tanımlanması

Freze kesicilerin "noktadan-noktaya hareket" ve "çevresel işleme hareketi" olmak üzere iki çeşit hareketi vardır. Noktadan-noktaya harekette, kesici takımın başlangıç noktası ve varış yeri arasında katettiği yol, bir düz çizgi olarak tanımlanır. Aksi takdirde, kesicinin katedeceği yol hakkında hiçbir bilgiye gerek olmadan sadece varış yerinin konumunu kontrol etmek gerekebilir. Bu nedenle; örneğin başlangıç konumundan gerekli derinliğe ulaşacak şekilde bir delik açılmasının noktadan noktaya bir doğrusal hareket olmasına karşın, matkabın bir dizi delme işleminin bir parçası olarak bir delikten başka bir deliğe konumlandırılması, kesici takımın hareket yolu üzerinde hiçbir özel bilgi gerektirmeyen bir noktadan-noktaya harekettir. Çevresel İşleme hareketi ise; kesici takımın, önceden belirlenmiş bir yol boyunca ve yoldan yaptığı sapmaları belirli bir tolerans içinde koruyarak gerçekleştirdiği bir harekettir. Bu hareket, istenen profilin elde edilmesini sağlar.

APT dili, ya APT'de ya da tanımlanmış geometrik şekillerin kullanımıyla NC işlemcisi tarafından hesaplanmış konum koordinatlarına dayanarak tanımlanabilen geometrik cisimleri esas almak suretiyle, bir kesici takım veya kesici hareket yolunu (yani, parçanın profilini) tanımlayabilmenizi mümkün kılar.

Profilin bir parçası olan bir geometrik cismin tanımlanması, kesici takımın istenilen konumunu ya da yolunu hesaplamaya veya tanımlamaya kıyasla genellikle daha kolaydır. Bunun nedeni; programcıların genelde sadece çizime bakarak bir kesicinin yolunu tanımlamak için gerekli olan bilgileri bulamamaları; fakat kesicinin yarıçapı için gerekli telafiye sağlamak ve kesicinin yaptığı sapmaların belirli tolerans sınırları içinde kaldığından emin olmak için, çeşitli doğrultularda gerek bir geometrik cismin bir diğeriyle kesişimi ve gerekse gerekli kesici farkı (offset) hesaplamaları gerekir. Yine de, doğru bir tasarım resmi, her zaman için bir parça profilinin çeşitli kısımlarını tanımlamak için gerekli boyutları ve parametre-

leri sağlar. Bu nedenle, kesici yolunun parça profili esasına göre tanımlanması, NC işlemcisinin yolun tanımlanması için gerekli olan konum koordinatları, eksen kaçıklıkları, telafi gibi parametrelerin hepsini otomatik olarak hesaplayan yordamları da yapısında toplamış olduğunu ifade eder.

5.1 NOKTADAN-NOKTAYA HAREKETİN TANIMLANMASI

Noktadan noktaya hareket, adından da anlaşıldığı gibi, kesici takımın bir noktadan bir diğerine konumlanmasından oluşur. Noktadan noktaya hareketin belli türlerinde, hareket esnasında bir kesme işlemi istenmez. Kesici takımın herhangi bir konumdan daha güvenli bir konuma çekilmesi ve yeni bir kesme işlemine başlamak için hazır olacak şekilde kesicinin hızla bir konumdan diğerine geçmesi için gerekli olan hareketler, tipik örneklerdir. Hareket esnasında hiçbir kesme işlemi gerçekleşmez. Bu yüzden; belki tezgaha, iş parçasına, bağlama aparatına yapılabilecek müdahaleler dışında; kesici takımın hareket yolu ile ilgili olarak hiçbir şey yapmaya gerek yoktur. Diğer taraftan; bir kesici takımın, başlangıç noktasından varış noktasına kadar bir düz çizgi boyunca hareket etmesini gerekli kılan durumlar da mevcuttur.

APT dilinde, noktadan noktaya hareket, ya GOTO ya da GODLTA deyimleriyle tanımlanır. GOTO deyiminin formatı:

$$\text{GOTO/} \begin{Bmatrix} x,y,z,f \\ x,y \\ P1, f \end{Bmatrix}$$

olup burada;

x,y,z = varış noktasının koordinatları

P1 = önceden tanımlanmış ise, varış yerinin sembolü

f = bu hareket için gereken (ilerleme) hızı

(programda kullanılan birimlere bağlı olarak, dakikada inç veya milimetre cinsinden) belirtilen isteğe bağlı bir elemandır.

Bir GOTO deyimini belirtildiğinde, kesici takım bir önceki deyim tarafından tanımlanmış olan konumundan yeni belirtilen konumuna hareket eder; kesici takım ekseninin Z eksenine paralel olduğu varsayılır. Z koordinatı, eğer deyimden atılmışsa, sıfır olarak kabul edilir.

GODLTA deyiminin formatı ise:

$$\text{GODLTA/} \begin{Bmatrix} x,y,z \\ z \\ v1 \end{Bmatrix} [f]$$

olup burada;

x,y,z = Bir önceki deyimle tanımlanmış olan başlangıç noktasından varış noktasına kadar olan hareket yolu üzerinde, sırasıyla x,y ve z doğrultularındaki artışı uzaklıklardır. (Sadece z koordinatı belirtildiği takdirde, x ve y sıfır olarak kabul edilir ve kesici takım yalnızca Z yönünde hareket eder.)

V1 = Noktadan-noktaya hareketin yönünü ve büyüklüğünü belirten tanımlanmış bir vektördür.

f = Bu hareket için gerekli ilerleme hızıdır.

Bu iki deyim, bir noktadan noktaya hareketin gerçek yolunu tanımlamaz. Ayrıca; yukarıda belirtildiği gibi iki tip harekete gerek duyarız. Bir noktadan noktaya hareket yolunun düz bir çizgi olması gerektiğinde, genellikle bir kesme işlemi de vardır ve kesici takım belirtilen ilerleme hızında hareket etmelidir. Diğer taraftan, bir noktadan noktaya hareket yolunun kesme işlemi ile hiçbir ilgisi yoksa, çoğunlukla yüksek hızda bir hareket gösterilir. Bu iki hareket, bir NC programında G00 ve G01 kodlarıyla belirtilir. Bu yüzden bir GOTO ya da GODLTA deyimini tarafından tanımlanan hareket, belirtilen ilerleme hızıyla kontrol edilebilir. Bir noktadan noktaya hareket deyimini işlendiğinde; NC işlemcisi, belirtilen ilerleme hızı ve varış noktasını CLDATA dosyasında çıktı olarak yazar. Daha sonra bu değerler, bir son işlemci tarafından, eğer yüksek hızda bir hareket belirtilmişse bir G00 koduna, yüksek hız dışında bir ilerleme hızı belirtilmiş olduğu takdirde de bir G01 koduna dönüştürülür. Eğer bir noktadan noktaya hareket deyimini, kesici takım konumundaki değişikliği sadece bir koordinat ekseninde belirtirse, hareket daima doğrusaldır.

Hızlı hareket değeri, ilerleme hızı seçeneği olmaksızın GOTO veya GODLTA noktadan noktaya hareket deyiminden hemen önce belirtilmesi gereken

RAPID

deyimleriyle tanımlanır. Yüksek hız dışındaki bir ilerleme hızı, isteğe bağlı "f" elemanını belirtmek suretiyle GOTO ya da GODLTA deyiminde (dakikada inç veya milimetre cinsinden) belirtilebilir veya GOTO, GODLTA, GO veya FROM gibi önceki bir deyim içinde, ya da

FEDRAT/f

biçimindeki ayrı bir ilerleme hızı deyiminde tanımlanabilir. Burada; f , istenen ilerleme hızıdır. RAPID deyiminin sadece bir sonraki hareket deyimine kadar etkili

olduğuna da dikkat etmek gerekir. Onunla beraber, bir APT programında belirtilen ilerleme hızı, yüksek hız dışında yani bir ilerleme hızı belirten bir deyim tarafından değiştirilene kadar etkinliğini korur.*

Programın başında, henüz kesici takımın konumu tanımlanmamışken kesicinin ilk konumunu tanımlayabilmek için bir FROM deyimi belirtilmelidir. Bu deyim formatı:

$$\text{FROM} / \left\{ \begin{array}{l} P1[,A] \\ x,y,z[,A] \\ x,y \end{array} \right\}$$

olup burada; P1 tanımlanmış bir noktanın sembolü; x, y ve z ise bu noktanın koordinatlarıdır. Eğer üçüncü seçenek tercih edilirse, z koordinatı sıfır olarak alınır.

Şekil 2-21'de gösterilen parça için yapılacak delme işlemi, APT'de noktadan noktaya hareket deyiminin tipik bir uygulamasını temin eder. Aşağıda, parça üzerinde 12 adet delik açmak için kullanılan programın bir bölümü gösterilmiştir.

```

.....
.....
FROM/-4,7,3.0    $$$BAŞLANGIÇ NOKTASI (-4,7,3)
RAPID;GOTO/0,0,3    $$$KESİCİ TAKIMI HIZLA (0,0,3) NOKTASINA GETİR
RAPID;GODLTA/-2.9 $$$ KESİCİ TAKIMI HIZLA (0,0,0.1) NAKTASINA İNDİR
FEDRAT/3.0;GODLTA/-1.0    $$$DELİĞİ DEL
RAPID;GODLTA/1.0    $$$HIZLA YUKARI ÇIK
RAPID;GODLTA/4,0,0    $$$HIZLA (4,0,0.1) NOKTASINA GİT
GODLTA/-1.0    $$$DELİĞİ DEL. İLERLEME HIZI ÖNCEKİ DEYİMDEN
                    $$$ 3.0 OLARAK ALINIR
RAPID;GODLTA/1.0    $$$HIZLA YUKARI ÇIK
RAPID;GODLTA/4,0,0    $$$HIZLA BİR SONRAKİ DELİK KONUMUNA GİT
.....
.....

```

5.2 BİR NOKTALAR DİZİSİ İÇİN NOKTADAN NOKTAYA HAREKETİN TANIMLANMASI: PATERN DEYİMİ

Mühendislik tasarımında delikler, bir parçayı oluşturan temel öğelerin arasında yer alır. Delikler simetrik ya da asimetrik olarak kendilerine özgün bir dizi içinde dağılmışlardır. Yukarıda verilen örnekte görüldüğü gibi, delik sayısının çok olması birçok noktadan noktaya hareket deyimini gerektirir. Her nokta için üç deyim gereklidir: biri kesici takımın belirli bir noktaya gitmesini, diğer ikisi ise takımın aşağı ve yukarı hareket etmesini sağlar. APT dili, bir noktalar kümesini bir dizi

*Bir ilerleme hızının kendisinden sonra gelen hareket deyimlerinin biri veya tümü için geçerli olduğu durum, kullanılacak son işlemcinin tasarımına bağlıdır. Burada açıklanmış olduğumuz hususlar, tipik bir tasarımı yansıtmaktadır. Daha ayrıntılı bilgi için, bu kitabın Kısım IV'e bakınız.

olarak tanımlayabilmemizi mümkün kılar. Böylece, kesicinin başlangıç noktasından dizideki çeşitli noktalara kadar olan hareketlerini tanımlamak için tek bir GOTO ifadesi kullanılır.

Rastgele dağılmış ya da aralarında geometrik bir ilişki bulunan bir noktalar kümesi,

PATERN/.....

biçimindeki bir dizi tanımlama deyimi kullanılarak tanımlanabilir.

PATERN deyimi için dört değişik format tipi mevcut olup; bunlar aşağıda açıklanmış gibidir:

1. Daha önceden tanımlanmış noktaların düzenli bir listesini bir dizi olarak tanımlayan format (Tablo 5-1'deki 1 No.lu deyim formatı)
2. Koordinatlarıyla belirtilmiş olan noktaların düzenli bir listesini bir dizi olarak tanımlayan format (Tablo 5-1'deki 2 No.lu deyim formatı)
3. X-Y düzlemindeki bir çember ya da çemberler üzerinde bulunan noktaların düzenli bir listesini bir dizi olarak tanımlayan format (Tablo 5-1'deki 3 No.lu deyim formatı)
4. Bir doğru ya da doğrular üzerinde bulunan noktaların düzenli bir listesini bir dizi olarak tanımlayan format (Tablo 5-1'deki 4 No.lu deyim formatı)

Bu formatlara ait örnekler de Tablo 5-1'de verilmiştir.

Bir dizi, dizideki noktaları belirtmek için en uygun koordinat sistemini seçerek tanımlanabilir. Dizi bir kez tanımlandıktan sonra, daha ileri düzeyde manipülasyon yapabilmek için, bazı parametrelerle birlikte aşağıda belirtilen değiştiricilerden biri veya birkaçı kullanılabilir:

TABLO 5-1 BİR NOKTALAR KÜMESİNİ DİZİ OLARAK TANIMLAYAN DEYİMLER

No.	Bilinen Durum	Deyim Formatı
1	Dizi P1, P2, ..., Pn şeklinde düzenli listelenmiş olarak verilen noktalardan oluşur.	PATERN/P1,P2,...,Pn

Yorum P1'den Pn'ye kadar olan sembollerden herhangi biri yerine, bir içiçe nokta tanımlama deyimini kullanılabılır. Bu deyimdeki noktaların sırası, kesici takımın hareket yolunu ve bir noktadan diğerine yaptığı hareketin sırasını verir.

Örnek

PATERN/P1,(P2=POINT/0,2,4),(POINT/3,7,5),(1,0,0)

deyimini dört noktayı bir dizi olarak tanımlamaktadır. Birinci nokta (P1), önceki bir deyimde tanımlanmıştır.

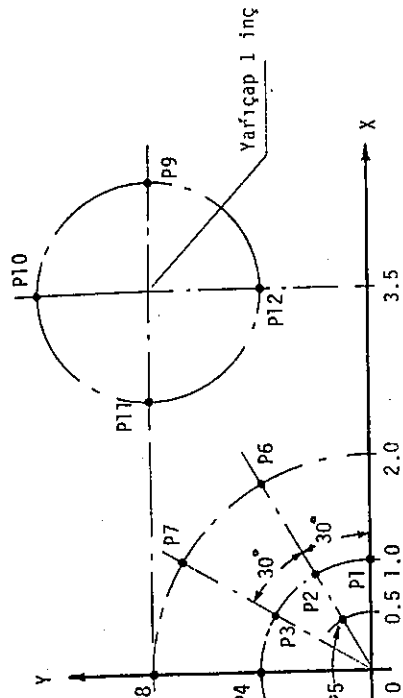
- a. Bu dizide listelenmiş olan noktalar, kutupsal merkezi P1 (x,y,z) noktasında olacak şekilde, $i = 1, 2, \dots, n$ olmak üzere bir dizi kutupsal koordinat (ri, ai) vasıtasıyla tanımlanmıştır. Kutupsal uzaklıklar için kullanılan semboller; r_i ($i = 1, 2, \dots, n$); açılar için olanlar ise a_i ($i = 1, 2, \dots, n$)'dir. Eğer kutupsal merkezin z koordinatı belirtilmemişse, tüm noktalar X-Y düzleminde tanımlanırlar.

	$\text{PATERN} \left[\begin{matrix} P1 \\ AT, \{x,y,z\} \end{matrix} \right],$	$\left\{ \begin{matrix} a_1, r_1, \dots, a_n, r_n \\ \text{CONST}, a_1, r_1, r_2, \dots, r_m \\ a_1, r_1, r_2, \text{CONST}, a_3, r_3, r_4, \dots, r_m \end{matrix} \right\}$	$\text{THETA},$
		$\left\{ \begin{matrix} r_1, a_1, \dots, r_m, a_n \\ \text{CONST}, r_1, a_1, a_2, \dots, a_m \\ r_1, a_1, a_2, \text{CONST}, r_3, a_3, a_4, \dots, a_n \end{matrix} \right\}$	$\text{RTHETA},$

Yorum

- Eğer belirtilmişse, P1 kutupsal merkez noktası, tanımlanan diziyeye dahil değildir.
- Kesme işaretinden hemen sonra gelen ve kutupsal merkezi tanımlayan iki eleman atılırsa, koordinat sisteminin başlangıç noktası kutupsal merkez olarak kabul edilir. Kutupsal merkez sadece x ve y koordinatları belirtildiği takdirde, z koordinatı sıfır olarak kabul edilir.

- Her bir noktanın z koordinatı kutupsal merkezin z koordinatıyla belirtilir. Tanımlanan dizi tüm noktaları parantez olan düzlemdir.
- CONST sözcüğü ikinci bir CONST sözcüğü kendisini ipal edene kadar, kendisinden hemen sonra gelen parametrenin dizi içinde kendisinden sonraki tüm noktalar için geçerli olduğunu gösterir.



Örnek

- (1) PT1 = PATERN/THETA,CONST,1,0,30,60,90
deyimini, hepsi de X-Y düzleminde bulunan P1,P2,P3 ve P4 noktalarından oluşan bir diziyi tanımlar.
- (2) PT2 = PATERN/AT,0,0,1,RTHETA,CONST,1,0,30,60,90
deyimini, hepsi de z = 1 düzleminde bulunan P1,P2,P3 ve P4 noktalarından oluşan bir diziyi tanımlar.
- (3) PT3 = PATERN/THETA,0,5,30,CONST,1,0,30,60,90,CONST,2,30,60,90
deyimini, hepsi de X-Y düzleminde bulunan P5,P1,P2,P3,P4,P6,P7 ve P8 noktalarından oluşan bir diziyi tanımlar.
- (4) PT4 = PATERN/AT,3,5,2,0,3,THETA,0,1,180,1,90,1,270,1
deyimini, hepside belirttikleri sırayla z = 3 düzleminde bulunan P11, P10 ve P12 noktalarından oluşan bir diziyi tanımlar.

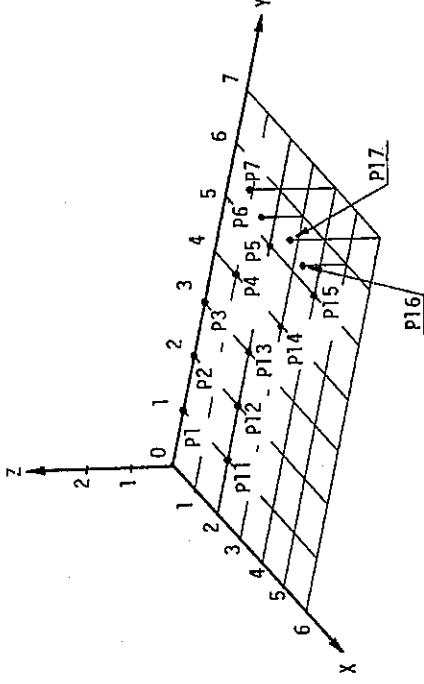
TABLO 5-1 BİR NOKTALAR KÜMESİNİ BİR DİZİ OLARAK TANIMLAYAN DEYİMLER (Devam)

No.	Bilinen Durum	Deyim Formatı
b.	Eğer DELTA seçeneği seçilmiş ise, dizide listelenen noktalar, bir noktadan bir sonraki artış Kartezyen koordinatları $(x_1, y_1, z_1, \dots, x_n, y_n, z_n)$ vasıtasıyla tanımlanmıştır. Bu dizideki birinci nokta, koordinatları (x, y, z) ya da, eğer z sıfır ise, (x, y) olan verilen bir referans noktasına göre tanımlanmıştır. ABSLTE seçeneği; $x_1, y_1, z_1, \dots, x_n, y_n, z_n$ 'nin dizideki noktaların mutlak koordinatları olduğunu gösterir. XYZ'nin dışındaki seçenekler için, her bir noktanın ihmal edilen koordinatları ya dizide bulunan bir önceki noktanın koordinatları ile ya da, kelime ilk nokta grubundan bir önce belirtilmişse, P1 noktasının koordinatları ile aynıdır.	$\text{PATTERN/} \left[\begin{array}{l} \text{AT, [P1]} \\ \left\{ \begin{array}{l} x, y, z \\ x, y \end{array} \right\} \end{array} \right] \left\{ \begin{array}{l} \text{DELTA,} \\ \text{[ABSLTE, J]} \end{array} \right\} \left\{ \begin{array}{l} \text{XYZ, } x_1, y_1, z_1, \dots, x_n, y_n, z_n \\ \text{NOX, } y_1, z_1, \dots, y_n, z_n \\ \text{NOY, } x_1, z_1, \dots, x_n, z_n \\ \text{INOZ, } x_1, y_1, \dots, x_n, y_n \\ \text{XCOORD, } x_1, x_2, \dots, x_n \\ \text{YCOORD, } y_1, y_2, \dots, y_n \\ \text{ZCOORD, } z_1, z_2, \dots, z_n \end{array} \right\} \$$ $\left[\begin{array}{l} \text{XYZ, } x_1, y_1, z_1, \dots, x_m, y_m, z_m \\ \text{NOX, } y_1, z_1, \dots, y_m, z_m \\ \text{NOY, } x_1, z_1, \dots, x_m, z_m \\ \text{NOZ, } x_1, y_1, \dots, x_m, y_m \\ \text{XCOORD, } x_1, x_2, \dots, x_m \\ \text{YCOORD, } y_1, y_2, \dots, y_m \\ \text{ZCOORD, } z_1, z_2, \dots, z_m \end{array} \right]$

Yorum

- (1) Eğer belirtilmişse, referans noktası P1 bu diziyeye dahil değildir.
- (2) Eğer referans noktası belirtilmemişse, koordinat sisteminin başlangıç noktası referans noktasıdır.
- (3) DELTA ve ABSLTE kelimeleri, kullanılan koordinatların sırasıyla artış ve mutlak koordinatlar olduklarını gösterirler.
- (4) XYZ, NOX, ..., veya ZCOORD kelimesi, dizide bulunan noktalar için, sonraki koordinatların hangi eksenlerde kullanıldığını gösterir. XYZ kelimesi; x, y ve z koordinatlarının her bir nokta için belirtildiğini gösterir. NOX, NOY veya NOZ kelimesinden sonra gelen her bir nokta için koordinatlar; sırasıyla y ile z, x ve z ve x ve y'dir. XCOORD, YCOORD ya da ZCOORD kelimesi; kendisinden sonraki noktalar için sadece x, y ya da z koordinatının verilmiş olduğunu gösterir.

(5) ABSLTE öndeger seçeneği olup; ihmal edilebilir. Yine bir öndeger seçeneği olan NOZ ise, ilk koordinat grubundan önce belirtildiği takdirde, ihmal edilebilir.

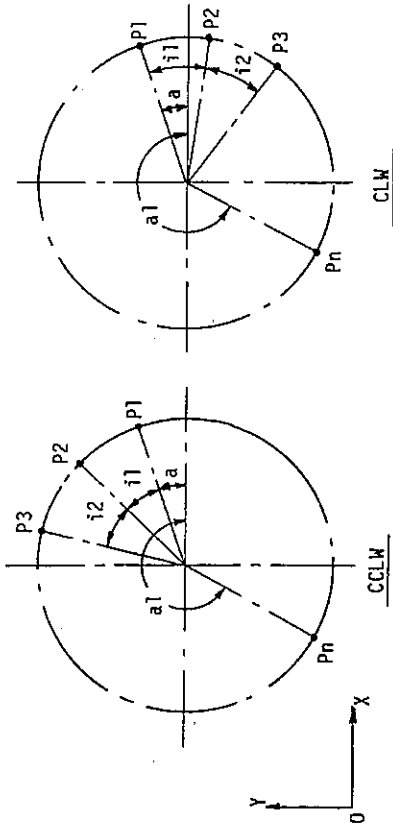


Örnek

- (1) $\text{P1} = \text{PATTERN/DELTA,YCOORD,1,1,1,NOZ,1,1,1,XYZ,1,1,1,1,1,1}$ deyimi, P1'den P7'ye kadar (dahil) olan noktaları kapsayan bir nokta dizisini tanımlar.
- (2) $\text{P2} = \text{PATTERN/VAT,2,0,0,DELTA,YCOORD,1,1,1,NOZ,1,1,1,XYZ,1,1,1,1,1,1}$ deyimi, P11'den P17'ye kadar (dahil) olan noktaları kapsayan bir nokta dizisini tanımlar. Bu iki örnekte, NOZ sözcüğünün ihmal edilemeyeceğine dikkat ediniz.
- (3) $\text{PATTERN/VAT,0,1,0,ABSLETE,YCOORD,1,2,3,NOZ,1,4,2,5,XYZ,3,6,1,4,7,2}$ deyimi, yukarıdaki Örnek 1 ile aynı diziyi tanımlar.

TABLO 5-1 BİR NOKTALAR KÜMESİNİ BİR DİZİ OLARAK TANIMLAYAN DEYİMLER (Devam)

No.	Bilinen Durum	Deyim Formatı
3	Tanımlanan dizideki noktalar kümesi; X-Y düzleminde bulunan ve merkezi (x,y) noktasındaki; r yarıçaplı C1 çemberi üzerindedir. İlk nokta a, son nokta ise a1 açısıl konumundadır. Eğer OMIT veya RETAIN sözcüğü belirtilmemişse, bu dizideki nokta sayısı n'dir.	$\text{PATTERN/ARC, } \left\{ \begin{array}{l} \text{C1} \\ \text{x,y,r} \end{array} \right\}, \text{ATANGL,a, } \left\{ \begin{array}{l} \text{CLW} \\ \text{CCLW} \end{array} \right\}, \left\{ \begin{array}{l} \text{a1,NUMPTS,n} \\ \text{INCR,i,NUMPTS,n} \\ \text{INCR,i1/2,...,(n-1)} \end{array} \right\}, \$$ $\left[\begin{array}{l} \text{[OMIT]} \\ \text{[RETAIN]} \end{array} \right], \text{n1,n2,...}$



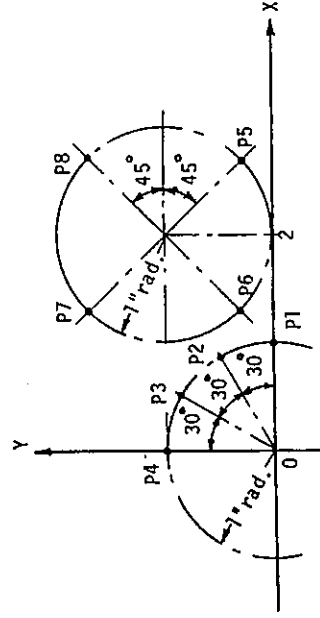
Yorum

- (1) a ve a1 açıları daima pozitif X yönünden itibaren ölçülürler. Noktaların ilkinden sonuncusuna doğru sıralandıkları yönü tanımlayan CLW ve CCLW sözcükleriyle hiçbir ilişkileri yoktur.
- (2) "a1,NUMPTS,n" seçeneği bu dizide n sayıda nokta olduğunu ve a ile a1 açısıl konumları arasında eşit aralıklı olarak yerleşiklerini gösterir.
- (3) "INCR,i,NUMPTS,n" seçeneği, bu dizide, ardışık iki nokta arasında aynı artışlı açı (i) bulunan n sayıda nokta olduğunu gösterir.

- (4) "INCR,i1,i2,...,(n-1)" seçeneği, bu dizide (m) nokta ile (m+1) nokta arasında birim açısıl açı bulunan n sayıda nokta olduğunu gösterir.
- (5) RETAIN sözcüğünün sadece (n1),(n2),... noktaların diziyeye dahil olması gerektiğini göstermesine karşın; OMIT seçeneği (n1),(n2),... noktaların diziden çıkarılması gerektiğini gösterir.

Örnekler

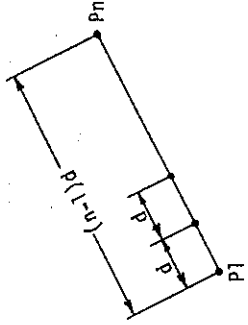
- (1) PT1 = PATTERN/ARC,0,0,1,ATANGL,0,CCLW,90,NUMPTS,4
deyimi vastasıyla tanımlanan dizideki noktaların sırası P1,P2,P3,P4'dür.
- (2) PT2 = PATTERN/ARC,2,1,1,ATANGL,-45,CLW,45,NUMPTS,4
deyimi vastasıyla tanımlanan dizideki noktaların sırası P5,P6,P7,P8'dir.



- (3) PT3 = PATTERN/ARC,0,1,ATANGL,0,CCLW,90,NUMPTS,4,RETAIN,1,3,4,ARC,\$
2,1,1,ATANGL,-45,CLW,45,NUMPTS,4,OMIT,2
deyimi vastasıyla tanımlanan dizideki noktaların sırası P1,P3,P4,P5,P7,P8'dir. Bu örnek aynı zamanda, kesme işaretinden sonraki bütün elemanlar kümesinin başka bir elemanlar kümesine sıra ile eklenebileceğini gösterir.

TABLO 5-1 BİR NOKTALAR KÜMESİNİ BİR DİZİ OLARAK TANIMLAYAN DEYİMLER (Devam)

No.	Bilinen Durum	Deyim Formatı
4 a.	Bu dizide tanımlanan P_1, \dots, P_n noktalar kümesi, ilk ve son noktalar (P_1 ve P_n) tarafından tanımlanmış olan bir düz çizgi üzerinde, noktalar arasında eşit uzaklık olacak şekilde bulunur.	PATERN/LINEAR,P1,Pn,NUMPTS,n



PATERN/LT/REAR,P1,P2, NUMPTS,n

(a)

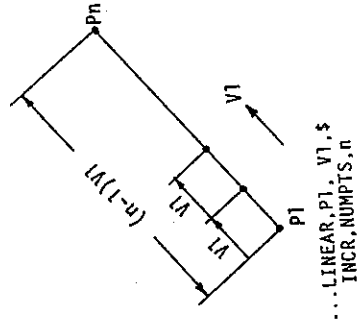
b. Bu dizide tanımlanan P_1, \dots, P_n noktalar kümesi, ilk noktası (P_1), bir eğim açısı (a) ya da bir V_1 vektörüyle tanımlanan düz bir doğru üzerindedir. P_2, \dots, P_n noktaları eşit d (NUMPTS seçeneği için) aralıklarıyla ya da belirtilen $d_1, d_2, \dots, d_{(n-1)}$ aralıklarıyla artan (INCR) ya da azalan (DECR) yönde yerleşebilirler. Eğer

$$\text{PATERN/LINEAR,P1,} \left\{ \begin{array}{l} \text{ATANGL,a} \\ \text{INCR} \\ \text{DECR} \end{array} \right\}, \left\{ \begin{array}{l} d, \text{NUMPTS,n} \\ d_1, d_2, \dots, d_{(n-1)} \end{array} \right\}, \left\{ \begin{array}{l} \text{OMIT} \\ \text{RETAIN} \end{array} \right\}, n_1, n_2, \dots$$

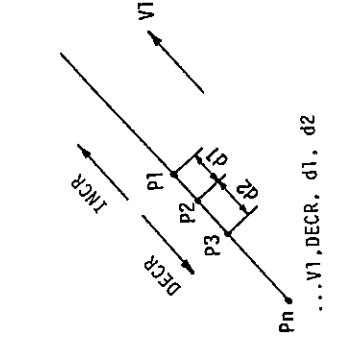
$$V_1, \left\{ \begin{array}{l} \text{INCR} \\ \text{DECR} \end{array} \right\}, d, \text{NUMPTS,(n-1)} \\ \left\{ \begin{array}{l} \text{INCR} \\ \text{DECR} \end{array} \right\}, d_1, d_2, \dots, d_{(n-1)}$$

$$\left\{ \begin{array}{l} \text{INCR} \\ \text{DECR} \end{array} \right\} d$$

seçeneği atılmış ve V_1 vektörü belirtilmiş ise; V_1 vektörü, bir nokta ile bir sonraki nokta arasında artışı vektör olarak kullanılır.



...LINEAR,P1,V1,\$ INCR,NUMPTS,n



...V1,DECR,d1,d2

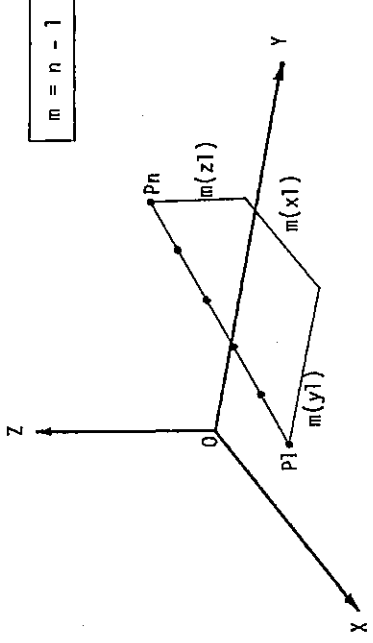
(b)

$$\text{PATERN/LINEAR,P1,DELTA,x1,y1,L2,NUMPTS,n-1,} \left\{ \begin{array}{l} \text{OMIT} \\ \text{RETAIN} \end{array} \right\}, n_1, n_2, \dots$$

c. Bu dizide tanımlanan n sayıdaki nokta (P_1, \dots, P_n) ilk noktadan (P_1) geçen ve bir noktadan bir sonrakine x_1, y_1 ve z_1 artışı koordinatları vasıtasıyla belirtilen yöndeki düz bir doğru üzerindedirler.

TABLO 5-1 BİR NOKTALAR KÜMESİNİ BİR DİZİ OLARAK TANIMLAYAN DEYİMLER (Devam)

No.	Bilinen Şart	Deyim Formatı
-----	--------------	---------------



(c)

Yorum $\left\{ \begin{array}{l} \text{OMIT} \\ \text{RETAIN} \end{array} \right\}$ n_1, n_2, \dots biçimindeki seçmeli elemanlar, evvelce açıklanan anlamların aynısını ifade eder.

Örnek P1, P2 ve P3 noktaları verilmişse, şu deyimler elde edilebilir:

PT1 = PATERN/INEAR,P1,P2,NUMPTS,4

PT2 = PATERN/INEAR,P2,ATANGL,0,INCR,0.5,NUMPTS,4,OMIT,1

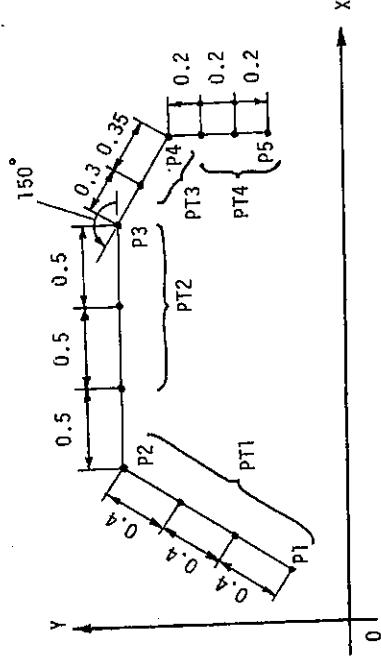
PT3 = PATERN/INEAR,P3,ATANGL,150,DECR,0.3,0.35,OMIT,1

PT4 = PATERN/INEAR,(POINT/PT3,2),DELTA,0,-0.2,NUMPTS,3,OMIT,1

Yukarıda şekilde gösterilen sırada P1'den P5'e kadar 12 nokta içeren dizi, yukarıda gösterilen tanımlama deyimlerindeki elemanların sırasıyla eklenmesiyle tanımlanabilir.

TABLO 5-1 BİR NOKTALAR KÜMESİNİ BİR DİZİ OLARAK TANIMLAYAN DEYİMLER (Devam)

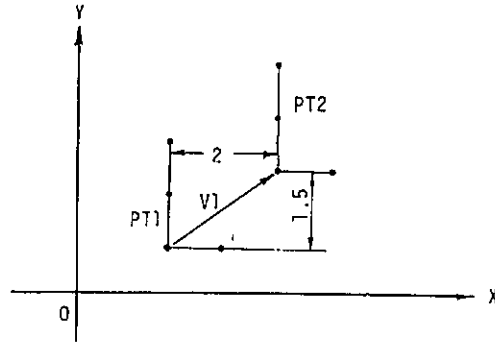
No.	Bilinen Şart	Deyim Formatı
-----	--------------	---------------



PT5=PATERN/INEAR, P1, P2, NUMPTS, 4, P3, NUMPTS, 4, ATANGL, 150, DECR, \$
0.3,0.35, DELTA, 0,-0.2, NUMPTS, 3

TABLO 5-2 BİR DİZİYİ İŞLETMEK İÇİN KULLANILAN DEĞİŞTİRİCİLER

Değiştirici	Deyim Formatı ve Açıklaması
TRANSL	PATERN/PT1,TRANSL, $\left\{ \begin{matrix} V1 \\ x,y, [z] \end{matrix} \right\}$



Yorum Dizi, verilen bir PT1 dizisinin bir V1 vektörüyle ya da x,y ve z artışlı koordinatlarıyla hareket ettirmek suretiyle tanımlanır. z koordinatı, eğer ihmal edilmişse, sıfırdır.

Örnekler

PT2 = PATERN/PT1,TRANSL,V1
 PT2 = PATERN/PT1,TRANSL,2,1.5,0
 PT2 = PATERN/PT1,TRANSL,2,1.5

ATTACH

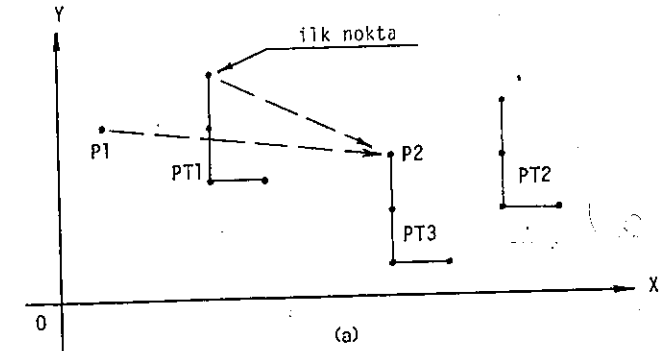
PATERN/PT1,[ATTACH,P1,] $\left\{ \begin{matrix} TRANPT,P2 \\ XYROT,a \\ INVX \\ INVY \end{matrix} \right\}$

Yorum

- Söz konusu dizi, verilen bir PT1 dizisinin belirtildiği biçimde dönüşümüyle tanımlanır. ATTACH sözcüğü, gelişigüzel seçilen bir P1 noktasının PT1, dizisine ilişitirildiğini ve bunların dönüşüm süresince yekpare bir gövde olarak dikkate alındığını gösterir. Aşağıdaki elemanlarla belirtilen bir dönüşüm, bu referans noktasına göre gerçekleştirilir. Eğer "ATTACH,P1" ihmal edilirse, referans noktası PT1 dizisindeki ilk noktadır.
- "TRANPT,P2" elemanı, PT1 dizisinin P1'den P2'ye hareket ettirilmesiyle elde edilen bir çeviriyi belirtir.

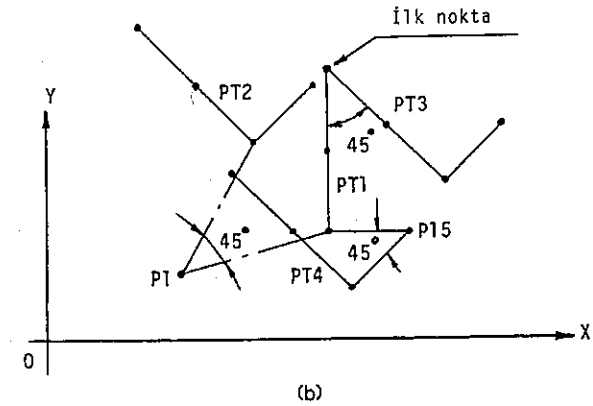
TABLO 5-2 BİR DİZİYİ İŞLETMEK İÇİN KULLANILAN DEĞİŞTİRİCİLER (devam)

Değiştirici	Deyim Formatı ve Açıklaması
-------------	-----------------------------

**Örnekler**

PT2 = PATERN/PT1,ATTACH,P1,TRANPT,P2
 PT3 = PATERN/PT1,TRANPT,P2

- "XYROT,a" elemanı, PT1 dizisinin P1 noktası etrafında a açısı kadar dönmelerini belirtir.

**Örnekler**

PT2 = PATERN/PT1,ATTACH,P1,XYROT,45
 PT3 = PATERN/PT1,XYROT,45
 PT4 = PATERN/PT1,ATTACH,P15,XYROT,45

- INVX ve INVY kelimeleri; verilen PT1 dizisinin eğer belirtilmişse P1 noktasından ya da eğer "ATTACH,P1" seçeneği ihmal edilmişse dizideki ilk noktadan geçen, ve sırasıyla (INVX) ve X (INVY) eksenlerine paralel olan bir doğru çevresinde değişimlerini belirtir.

TABLO 5-2 BİR DİZİYİ İŞLETMEK İÇİN KULLANILAN DEĞİŞTİRİCİLER (devam)

Değiştirici	Deyim Formatı ve Açıklaması
	<p>(c)</p>

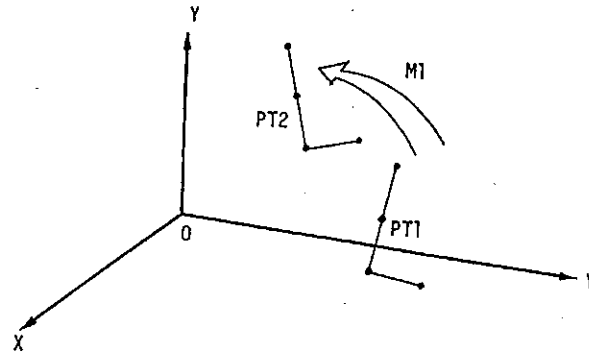
Örnek Eğer PT1, ilk noktası P11 olarak tanımlanan bir dizi ise, şu deyimler geçerlidir:

PT2 = PATTERN/PT1,ATTACH,P1,INVX

PT3 = PATTERN/PT1,ATTACH,P1,INVT

PT4 = PATTERN/PT1,INVX

MODIFY PATTERN/PT1,MODIFY,M1



Yorum Tanımlanan dizi, verilen PT1 dizisinden, verilen bir M1 matrisine* göre dönüştürülmüş bir dizedir.

Örnek Eğer PT1 dizisinin PT2'ye dönüşümü için gerekli olan matris M1 ise, PT2'yi tanımlayan deyim:

PT2 = PATTERN/PT1,MODIFY,M1

INVERS PATTERN/PT1,INVERS

biçimindedir.

* MATRIX deyiminin tanımı için, Bölüm 7'ye bakınız.

TABLO 5-2 BİR DİZİYİ İŞLETMEK İÇİN KULLANILAN DEĞİŞTİRİCİLER (devam)

Düzeltilici	Deyim Formatı ve Açıklaması
	<p>Yorum Bu dizide tanımlanan noktalar, PT1 dizisinde bulunanların aynıdır, fakat sıraları terstir.</p> <p>Örnek Eğer PT1 dizisi, listelenmiş sıralarıyla P1, P2 ve P3 noktalarından oluşuyorsa; aşağıdaki deyimle tanımlanan dizi; PT1 dizisindeki noktaların P3, P2, P1 olarak sıralanmasıyla oluşur.</p>
OMIT RETAIN	<p>PATTERN/PT1, { RETAIN OMIT RETAIN },n1,n2,...</p> <p>Yorum Bu iki değiştirici, Tablo 5-1'deki 3 ve 4 Nolu deyim formatlarında açıklananla aynı işleve sahiptir. OMIT değiştiricisi için, tanımlanan dizi, (n1), (n2), ... noktalar dışında PT1 dizisi ile aynı noktalar kümesine sahiptir. RETAIN değiştiricisi için; tanımlanan dizi PT1 dizisindeki noktalardan sadece (n1), (n2), ... noktaları içerir. Tanımlanan dizideki noktaların sırası PT1 dizisindeki sıranın aynıdır.</p>

Örnekler Üç nokta dizisi (PT2, PT3 ve PT4), aşağıdaki deyimler vasıtasıyla PT1 dizisi esas alınarak tanımlanmaktadır.

PT1 = PATTERN(0,0,0),(0,1,0),(0,2,0),(0,3,0),(0,4,0)

PT2 = PATTERN/PT1,OMIT,3

PT3 = PATTERN/PT1,RETAIN,2,4,5

PT4 = PATTERN/PT1,RETAIN,5,2,4

PT2 dizisi, listelenen sırada dört noktaya (0,0,0),(0,1,0),(0,3,0),(0,4,0) sahiptir. PT3 ve PT4 dizileri ise, birbirleriyle özdeş olup; aynı değerde ve sıradaki noktalar kümesine (0,1,0),(0,3,0),(0,4,0) sahip bulunur.

TRANSL Tanımlanmış bir diziyi döndürmeksizin bir noktadan diğerine hareket ettirir.

ATTACH Etrafında; çeviri (**TRANPT**), dönme (**XYROT**) veya değişim (**INVX** ya da **INYY**) gibi bir manipülasyonun yapıldığı bir noktayı gösterir.

MODIFY Manipülasyonun tanımlanan bir matrise göre gerçekleştiğini gösterir.

INVERS Dizideki noktaların sırasını ters çevirir.

OMIT Dizideki bazı noktaları atar.

RETAIN Dizideki bazı noktaları saklayıp diğerlerini geçersiz kılar.

Bu değiştiricilerin formatları ve kullanılışları Tablo 5-2'de sıra ile gösterilmiştir. Tablo 5-2'de listesi verilen bir veya daha fazla değiştirici bir **PATERN** deyiminde kullanılabilir. Böylece PT1 dizisi, şu bir tek deyimle sağlanan bir dizi dönüşümler vasıtasıyla işlenebilir:

PATERN/PT1,m1,m2,m3,...

Bu deyimde PT1, kesinlikle tanımlanmış bir diziyi ait sembol olmalıdır; ve m1,m2,... ise Tablo 5-2'de sıralanan manipülasyonların herhangi birisi olabilir. Manipülasyonların sırası, deyim içinde buldukları sıranın aynıdır. Manipülasyon sayısında hiçbir sınırlama yoktur. Örneğin Şekil 5-1 m1, m2 ve m3 sırayla "ATTACH,(0,0,0),INVX," "XYROT,-45," ve "ATTACH,P1,TRANPT,P2," ise; yukarıda gösterilen deyim; önce PT1 dizisinin Y eksenini etrafında değişimi, sonra dizideki ilk noktanın etrafında -45 derece döndürülmesi, son olarak da P1 noktasının P2 noktasına hareket ettirilmesi suretiyle oluşan bir diziyi tanımlar. Sonuç PT2 dizisidir. Bu değiştiricilerin bir uygulaması Şekil 5-2'de gösterilmiştir. İş parçası, her biri üç noktadan oluşan yedi tane noktalar kümesine sahiptir. Önce diziyi en uygun konum ve yöne yerleştirerek genel diziyi, sonra da uygun dönüşümler vasıtasıyla PT1'den PT7'ye kadar olan dizileri tanımlayabiliriz. Ortaya çıkan program aşağıdaki gibi olabilir:

.....
.....
PT0=PATERN/(0,1,0), (P0=POINT/0,0,0), (1,0,0)

PT1=PATERN/PT0,ATTACH,P0,XYROT,35,ATTACH,P0,TRANPT,(d1,d2,0)

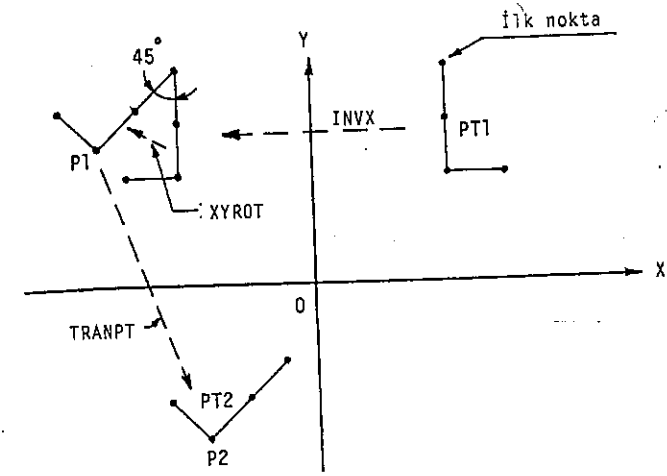
PT2=PATERN/PT1,ATTACH,P0,XYROT,60

PT3=PATERN/PT1,ATTACH,P0,XYROT,120

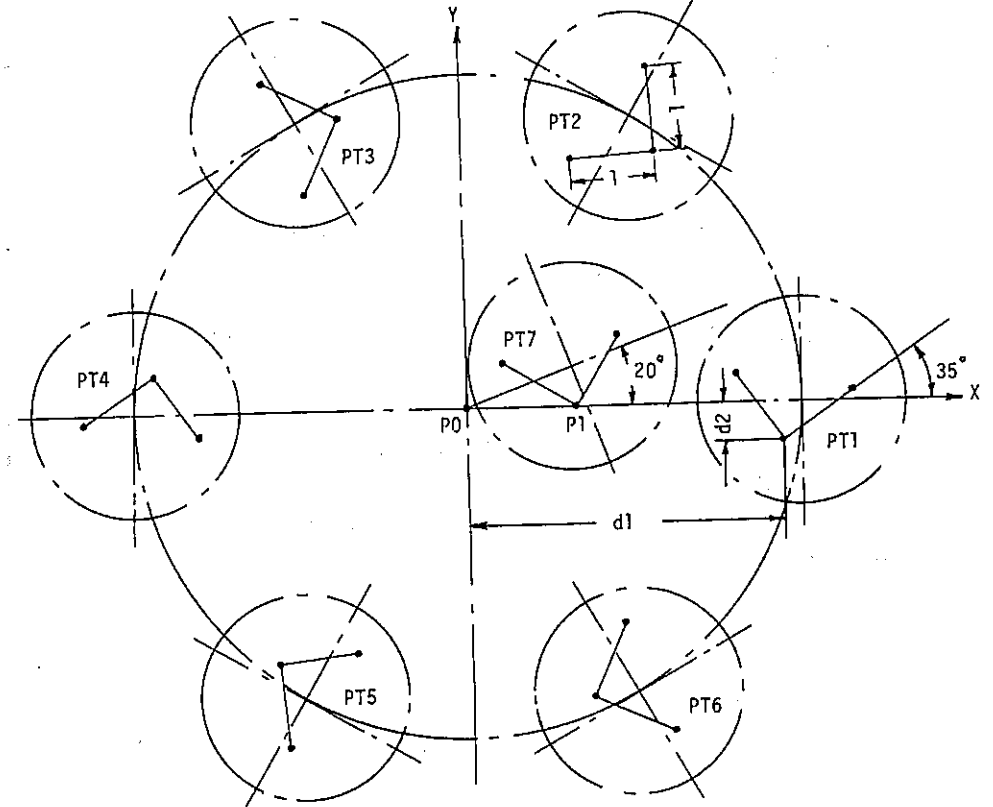
.....
.....
PT7=PATERN/PT0,ATTACH,P0,XYROT,55,ATTACH,P0,TRANPT,P1

.....

.....



Şekil 5-1 Çoklu dönüşümler vasıtasıyla oluşturulan bir dizi:
PT2=PATERN / PT1, ATTACH, (0,0,0), INVX, XYROT, -45, ATTACH, P1, TRANPT, P2



Şekil 5-2 Aynı diziyi sahip yedi adet noktalar kümesi.

NC işlemcisi de daha önce tanımlanmış birkaç diziyi yeni bir dizi oluşturmak için birleştirebilmemizi sağlar. Örneğin; PT1, PT2 ve PT3 tanımlanmış, üç dizi ise, bu diziler,

PT4=PATERN/PT1,PT3,PT2

deyimi kullanılarak yeni bir dizi oluşturmak için birleştirilebilirler.

Eğer PT1, PT2 ve PT3 dizilerinde tanımlanan noktaların sırası

P1, P2, P3, P4

P5, P6, P7

P8, P9, P10, P11

ise; bu noktaların yeni PT4 dizisindeki sıraları

P1, P2, P3, P4, P8, P9, P10, P11, P5, P6, P7 şeklindedir.

Bir tek diziyi işletmek için, yukarıdaki deyimde yeralan dizilerden birinin arkasından bir değiştirici de eklenebilir. Örneğin

PT5=PATERN/PT1,PT3,TRANPT,(1,0,0)

deyimi, PT1 dizisi ile, PT3 dizisinin +X yönünde bir birim hareket ettirilmesiyle, ortaya çıkan diziyi birleştirerek tek bir dizi (PT5) haline getirir.

PT6=PATERN/PT1,XYROT,90,PT3,TRANSL,1,0

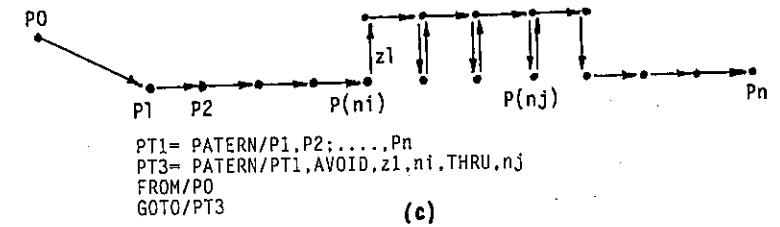
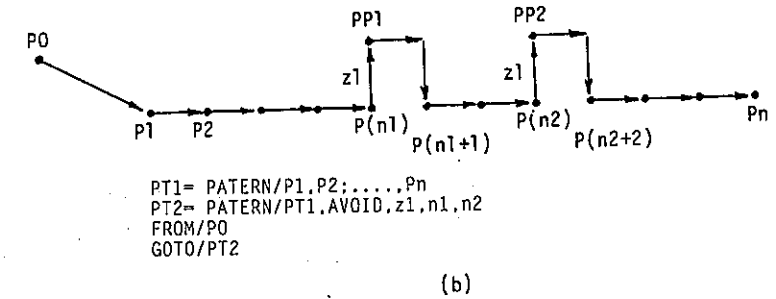
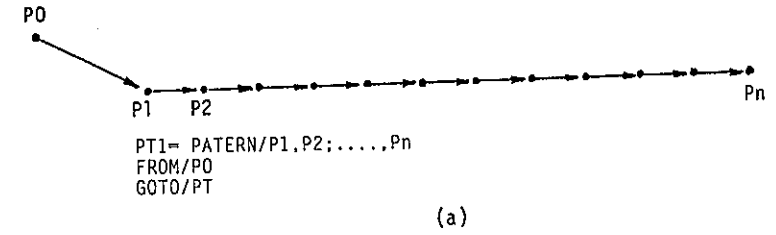
deyimi ise, PT1 dizisinin ilk noktası etrafında 90 derece döndürülmesiyle dönüşüm dizisi ile, PT3 dizisinin +X yönünde bir birim hareket ettirilmesiyle ortaya çıkan dönüşüm dizisini birleştirmek suretiyle bir tek dizi (PT6) haline getirir.

Kesici takımın bir dizide bulunan ve birbirini izleyen noktalara gitmesi için emir vermek amacıyla:

GOTO/PT1

formatlı bir GOTO deyimi kullanılır.

burada PT1, önceden tanımlanmış bir dizidir. Örneğin, eğer PT1 dizisi sırasıyla, P1, P2, P3 ve P4 şeklinde dört noktadan oluşuyorsa; yukarıdaki deyim, kesici takımını daha önceki bir deyim tarafından tanımlanmış ilk konumundan itibaren sırayla P1, P2, P3 ve P4 noktalarına yönlendirir. Bu deyim tek başına sıralı bir noktadan noktaya hareketi tanımlayabiliyorsa da; delme, kalıplama ya da deliğe diğ açma gibi gerçek bir işlemin programlanmasında güçlükle kullanılabilir. Bunun bir sebebi; kesme işleminin gerçekleştirilebilmesi için, kesici takımını dizideki her nokta üzerinde aşağıya ve yukarıya hareket ettirmek zorunda olmamızdan kaynaklan-



Şekil 5-3 AVOID değiştiricisinin anlamı. Şekiller (a) AVOID değiştiricisi olmadığında, (b) AVOID değiştiricisi olduğunda, (c) AVOID değiştiricisi ve THRU kelimesi olduğunda kesici takımın hareketini

maktadır. Ayrıca, dizide bulunan noktalar tek bir düzlem üzerinde olmayabilir ya da iki nokta arasında engeller (bir bağlama aparatı ya da parçanın kendisi) olabilir ve bu nedenle de kesici takımın bir noktadan müteakip noktaya direkt olarak gidebilmesi mümkün olmayabilir. Gerçek bir işlemi tanımlayabilmek için, ilave değiştiricilere gerek vardır.

5.3 BİR ENGELDEN SAKINMAK VEYA KAÇMAK İÇİN KULLANILAN İLAVE KESİCİ TAKIM HAREKETİNİ TANIMLAYAN DEĞİŞTİRİCİLER

Bir dizideki noktadan noktaya harekete ilave bir kesici takım hareketi dahil etmek için, bir PATTERN deyiminde, başlıca şu iki değiştirici kullanılabilir: AVOID (kesici takımı yükseltmek için) ve GOTO (kesici takımı bir engel etrafında hareket ettirmek için).

5.3.1 AVOID değiştiricisi

AVOID değiştiricisi, belli parametrelerle birlikte; tanımlanmış bir dizinin belli özel noktalarda kesici takımın yukarı kaldırılması gereken mesafeyi tanımlar. Bu deyim formatı şöyledir:

$$\text{PATTERN/PT1,AVOID, z1, } \left\{ \begin{array}{l} n_1, n_2, \dots \\ n_1, n_2, \dots, n_j \text{ THRU, } n_j, \dots \end{array} \right\}$$

burada; PT1 verilen bir dizidir; z1 parametresi ise PT1 dizisinin (n1), (n2), ... noktalarında kesici takımın yukarı kaldırılacağı yüksekliği belirtir. (Şekil 5-3 [b]) THRU sözcüğü; (ni) noktadan (nj) noktaya kadar ilave bir kesici takım hareketinin devreye girmesi gerektiğini gösterir. (Şekil 5-3[c]). AVOID değiştiricisine sahip olan ve olmayan dizilere ait kesici takım hareketlerindeki farklılık, şekillerde açıkça görülebilir. AVOID değiştiricisi yalnızca ilave bir kesici takım hareketi değil; aynı zamanda belirtilen her nokta için yeni bir nokta t üretir; yeni noktanın X ve Y koordinatları belirtilen noktalarınkilerle aynıdır; fakat z koordinatı ile z1'in toplamına eşittir.

Bu nedenle, her ne kadar NC işlemcisi PP1 ve PP2 ek noktaların P1, P2, ..., Pn noktalarında olduğu biçimde işlemden geçirmese de; Şekil 53(b)'deki PT2 dizisi P1, P2, ..., P(n1), PP1, P(n1+1), ..., P(n2), PP2, P(n2+1), ..., Pn olmak üzere n+2 adet noktaya sahiptir. Bu durum, bir deyim içinde birden fazla AVOID düzelticisi kullanıldığında dikkate alınmalıdır.

İçinde birden çok AVOID değiştiricisinde bulunan bir PATTERN ifadesinin formatı aşağıdaki gibidir:

$$\text{PATTERN/PT1,AVOID,z1,} \dots \text{[.AVOID,z2,} \dots \text{] [.AVOID,z3,} \dots \text{]}$$

İlk değiştiricinin verilen PT1 dizine, ikincisinin ise birinci tarafından düzeltilmiş olan diziyeye uygulandığına dikkat edilmelidir. Örneğin Şekil 5-4(c)'deki kesici takım hareketine gerek duyulduğunu (yani; kesici takımın birinci ve dördüncü noktalarda sırasıyla 1 inç ve 1.1 inç kadar yukarı kaldırılması gerektiğini) varsayalım. Bu durumda;

$$\text{GOTO/(PT4 = PATTERN/PT1,AVOID,1,1,AVOID,1,1,4)}$$

deyimi Şekil 5-4(d)'de gösterildiği gibi yanlış bir hareket türetir; çünkü PT1 dizisindeki dördüncü nokta, birinci AVOID tarafından düzeltilen dizinin beşinci noktasıdır. Doğru deyim

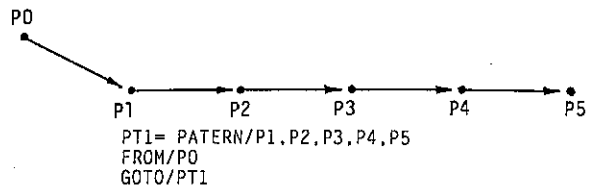
$$\text{GOTO/(PT3 = PATTERN/PT1,AVOID,1,1,AVOID,1,1,5)}$$

olması gerekir. (Şekil (5-4[c])).

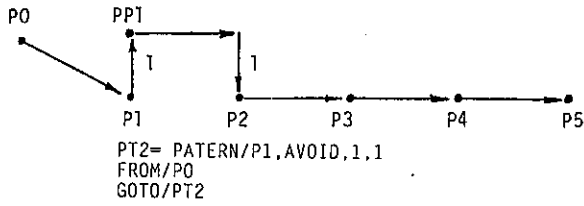
Eğer bir değiştiricide bulunan çeşitli noktaların z koordinatları eşit değilse, AVOID düzelticisi tarafından oluşturulan kesici takım hareketi de buna göre değişecektir (Şekil 5-5).

5.3.2 GOTO Değiştiricisi

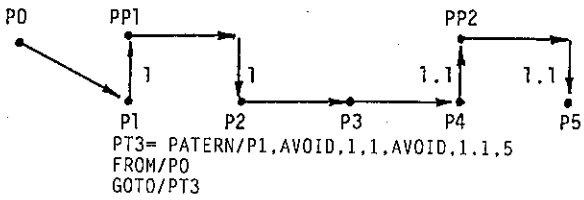
GOTO, bir PATTERN deyimindeki kesme işaretinden sonra belirtilmesi gereken bir alt kelime olup; daha önce geçen ve bir noktadan noktaya hareket, tanımlayan ana GOTO kelimesinden farklıdır. Bir PATTERN deyimindeki GOTO kelimesi, herhangi iki nokta arasındaki engeli atlatabilmek için kesici takımın uygun bir geri çekilme konumuna yapacağı ek bir hareketi belirtir. Bu yüzden iki noktanın sembolleri arasında bulunması gerekir. Örneğin Şekil 5-6'daki PT1 dizisi P1'den P5'e kadar beş noktadan oluşur. P3 ile P4 arasında bulunan engeli aşabilmek



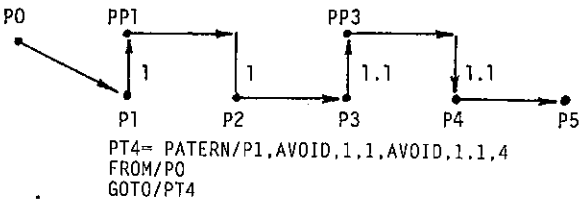
(a)



(b)

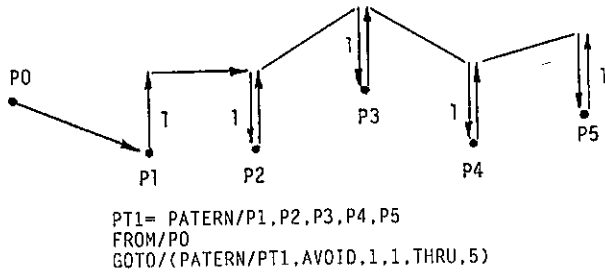


(c)

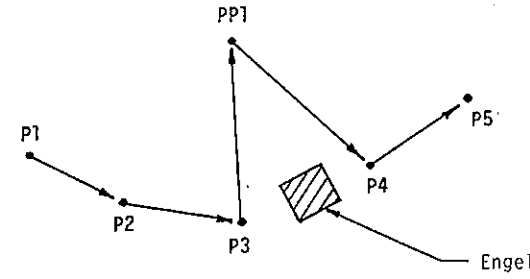


(d)

Şekil 5-4 Kademeli olarak eklenmiş AVOID değiştiricilerin kullanımı.



Şekil 5-5. AVOID değiştiricisinin kullanılmasıyla tanımlanan bir dizideki noktalar kümesine göre kesici takım hareketi.



Şekil 5-6. Bir engelin varlığı bir ilave kesici takım hareketini gerektirir.

İçin aşağıdaki deyim kullanılabilir:

PT1=PATERN/P1,P2,P3,GOTO,PP1,P4,P5

burada; PP1 uygun bir çekilme noktasıdır. Yine NC işlecisi bu noktaya dizideki diğer noktalarındaki işlemden geçirmez (Bölüm, 5.3.3'deki örneğe bakınız).

5.3.3 Bir Dizi Noktadan Noktaya Hareketin Programlanmasına Dair Örnekler

Şekil 2-21(a)'da görülen parça, dizi deyiminin AVOID değiştiricisi ile birlikte kullanılmasına dair tipik bir örnektir. 12 delik delmek için kullanılacak APT programı aşağıdaki gibi olabilir:

```

.....
.....
P0=POINT/ (-4,0,3)
PP0=POINT/ (0,0,0.1)
PT1=PATERN/ (0,0,-0.9),(4,0,-0.9),(8,0,-0.9),(12,0,-0.9),(12,3,-0.9),
(12,6,-0.9),(8,6,-0.9),(4,6,-0.9),(0,6,-0.9),(0,3,-0.9),$
(3,3,0.85),(9,3,0.85)
PT2=PATERN/PT1,AVOID,1,0,1,THRU,9,AVOID,3,0,19,AVOID,1,25,21,22
FROM/P0
GOTO/PP0
GOTO/PT2
.....
.....

```

Programın tamamı ve işleme sonucu Şekil 5-7'de gösterilmekte; bu program tarafından tanımlanan kesici takım hareketi ise Şekil 5-8'de verilmektedir. Program aynı zamanda, GOTO/(dizi) deyimini tarafından tanımlanan bir kesici takım hareke-

tinin kontrolü ve doğruluğunun kanıtlanması için kullanılan yöntemi de göstermektedir. Listeleme dosyasından görülebileceği gibi, daha önce kullanılan PRINT/3,.....

```

1
IBM S/370 APT-AC N/C PROGRAM VERSION=1.3.0000 DATE=07/28/86 TIME=12:43:58 0
... BEGIN TRANSLATION PHASE... (SECTION 1) ...
0
ISN 00001 PARTNO
ISN 00002 MACHIN/GNSCC,9,OPTION,2,0
ISN 00003 CLPRNT
ISN 00004 FEDRAT/2,0
ISN 00005 CUTTER/1,0
ISN 00006 PT1=PATERN/(0,0,-0.9),(4,0,-0.9),(8,0,-0.9),(12,0,-0.9),(12,3,-0.9),$
(12,6,-0.9),(8,6,-0.9),(4,6,-0.9),(0,6,-0.9),(0,3,-0.9),$
(3,3,0.85),(9,3,0.85)
ISN 00007 PT2=PATERN/PT1,AVOID,1,1,THRU,9,AVOID,3,19,AVOID,1.25,21,22
ISN 00008 PRINT/3,PT2

PT2 PATERN 24 ITEMS UNITS= INCHES

ISN 00009 FROM/-4,0,3
ISN 00010 GOTO/(0,0,0.1)
ISN 00011 GOTO/PT2
ISN 00012 END
ISN 00013 FINI

NO DIAGNOSTICS ELICITED DURING TRANSLATION PHASE
11 N/C SOURCE RECORDS (SYSIN)

```

```

SECTION 1 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.1433
SECTION 2 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.0733

```

```

1
....SECTION
0
ISN PARTNO/ MACHIN/ GNSCC 9.00000000 OPTION 2.00000000
0002
0004 FEDRAT/ 0.0 2.00000000
0005 CUTTER/ 1.00000000
0009 FROM/
0010 GOTO/ 0.0 0.0 0.0
0011 GOTO/ 0.0 0.0 0.1
0011 CYCLE / 0.0 0.0 -0.90000000
0011 GOTO/ NOMORE
0011 GOTO/ 0.0 0.0 0.10000000
0011 CYCLE / 4.00000000 0.0 0.10000000
0011 GOTO/ ON
0011 CYCLE / 4.00000000 0.0 -0.90000000
0011 GOTO/ NOMORE
0011 GOTO/ 4.00000000 0.0 0.10000000
0011 CYCLE / 8.00000000 0.0 0.10000000
0011 GOTO/ ON
0011 CYCLE / 8.00000000 0.0 -0.90000000
0011 GOTO/ NOMORE
0011 GOTO/ 8.00000000 0.0 0.10000000
0011 CYCLE / 12.00000000 0.0 0.10000000
0011 GOTO/ ON
0011 CYCLE / 12.00000000 0.0 -0.90000000
0011 GOTO/ NOMORE

```

Şekil 5-7 Bir APT program ve basılı çıktısı. Program, sıra ile devreye giren değiştiricilere sahip bir dizi deyiminden oluşmaktadır.

deyimi, dizide tanımlanan her bir noktanın tek tek çıktılarını vermez. Kesici takım hareketinin doğru olup olmadığının kanıtlanabilmesi için, birbirini izleyen kesici takım konumların, koordinatların listeleme dosyasında yazılabilesini sağlamak üzere, kesici konum verilerinin (CLDATA) basılmasını isteyen.

```

0011 GOTO/ 12.00000000 0.0 0.10000000
0011 CYCLE / 12.00000000 3.00000000 0.10000000
0011 GOTO/ ON
0011 CYCLE / 12.00000000 3.00000000 -0.90000000
0011 GOTO/ NOMORE
0011 CYCLE / 12.00000000 3.00000000 0.10000000
0011 GOTO/ 12.00000000 6.00000000 0.10000000
0011 CYCLE / 12.00000000 6.00000000 -0.90000000
0011 GOTO/ ON
0011 CYCLE / 12.00000000 6.00000000 0.10000000
0011 GOTO/ 12.00000000 6.00000000 0.10000000
0011 CYCLE / 8.00000000 6.00000000 0.10000000
0011 GOTO/ ON
0011 CYCLE / 8.00000000 6.00000000 -0.90000000
0011 GOTO/ NOMORE
0011 CYCLE / 8.00000000 6.00000000 0.10000000
0011 GOTO/ 8.00000000 6.00000000 0.10000000
0011 CYCLE / 4.00000000 6.00000000 0.10000000
0011 GOTO/ ON
0011 CYCLE / 4.00000000 6.00000000 -0.90000000
0011 GOTO/ NOMORE
0011 CYCLE / 4.00000000 6.00000000 0.10000000
0011 GOTO/ 4.00000000 6.00000000 0.10000000
0011 CYCLE / 0.0 6.00000000 0.10000000
0011 GOTO/ ON
0011 CYCLE / 0.0 6.00000000 -0.90000000
0011 GOTO/ NOMORE
0011 CYCLE / 0.0 6.00000000 0.10000000
0011 GOTO/ 0.0 3.00000000 0.10000000
0011 CYCLE / 0.0 3.00000000 -0.90000000
0011 GOTO/ ON
0011 CYCLE / 0.0 3.00000000 2.10000000
0011 GOTO/ 3.00000000 3.00000000 3.85000000
0011 CYCLE / 3.00000000 3.00000000 0.85000000
0011 GOTO/ ON
0011 CYCLE / 3.00000000 3.00000000 2.10000000
0011 GOTO/ 9.00000000 3.00000000 2.10000000
0011 CYCLE / 9.00000000 3.00000000 0.85000000
0011 GOTO/ ON
0011 CYCLE / 9.00000000 3.00000000 2.10000000
0011 GOTO/ 9.00000000 3.00000000 2.10000000
0012 END
0013 ***** FINI *****
.....END OF SECTION 3....

```

SECTION 3 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.1299

Şekil 5-7 (Devam)

CLPRINT

deyimi kullanılabilir.

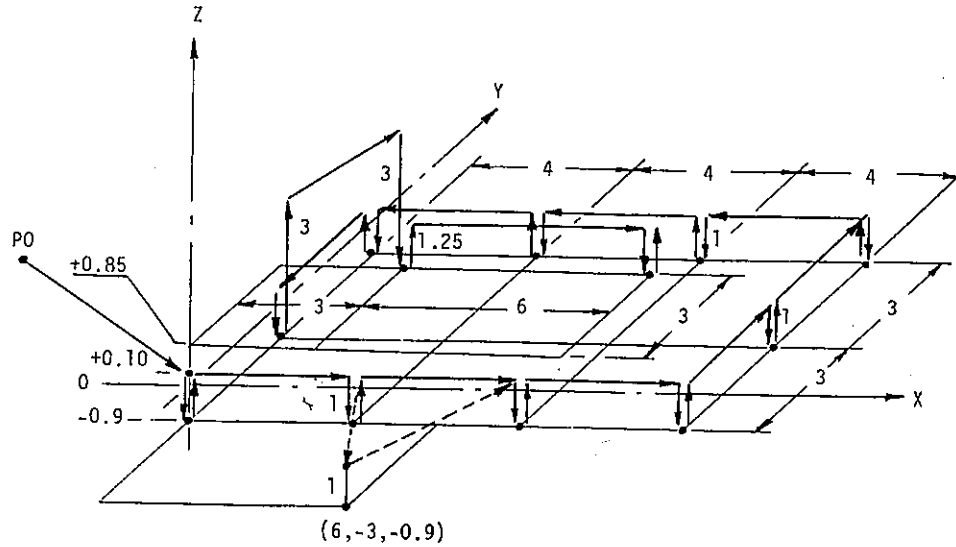
Eğer PT1 dizisinin ikinci ve üçüncü noktaları arasında bir engel varsa, programdaki (Şekil 5-7) deyimi:

PT1=PATERN/(0,0,-0.9),(4,0,-0.9),GOTO,(6,-3,-0.9),(8,0,-0.9),...

biçiminde değiştirebilir ve programı tekrar çalıştırabiliriz. Şekil 5-8'de, kesici takımın ikinci ve üçüncü noktalar arasındaki hareketi kesik çizgiyle belirtilmiş olup; bu da delme işleminin emniyetli bir geri çekme noktasında yapılmadığını ve AVOID değiştiricisi için belirtilen z koordinatının emniyetli geri çekme noktasının z koordinatlarına ilave edilmiş olduğunu gösterir.

5.4 ÇEVRESEL İŞLEME HAREKETİNİN TANIMLANMASI

Çevresel işleme hareketi, kesicinin, tanımlanmış olan profilden yapacağı sapmaların daima belirtilen bir tolerans sınırı içinde korunarak parça profili boyunca yaptığı harekettir. Sonuç olarak, belirtilen belli biçimde bir parça üretilebilir.



Şekil 5-8 Şekil 5-7'de verilen programın tanımlandığı kesici takım hareketi. Kesik çizgiler GOTO değiştiricisi tarafından tanımlanan ilave kesici hareketlerini göstermektedir.

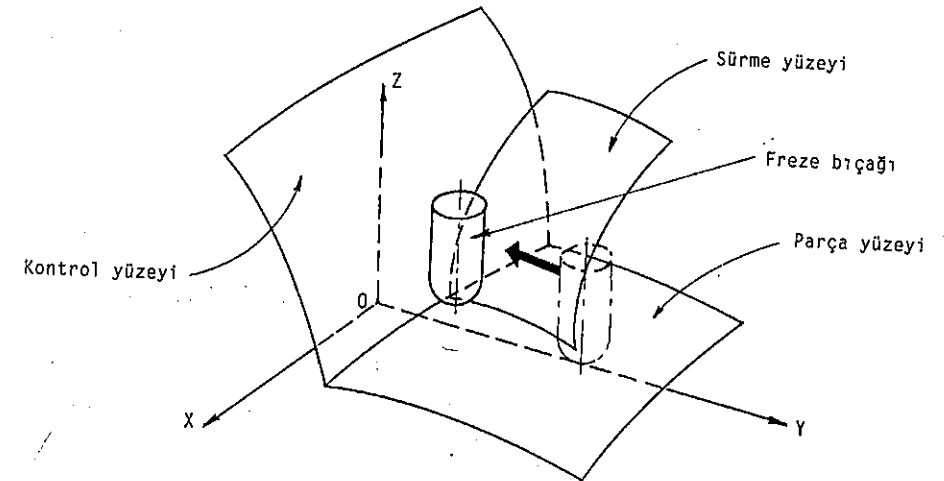
APT programlamanın ayırıcı özelliklerinden birisi, kesicinin çevresel işleme hareketinin tanımlanmasında sağladığı göreceli kolaylıktır. Çevresel işleme hareketinin NC programlaması sırasında başlangıç ve bitiş noktaları koordinatları-

nın hesaplanması ve belirtilmesi gerekir; ve alet hareketi yalnızca doğrusal ya da dairesel olarak tanımlanabilir. Eğer bir parçanın profili doğrusal ya da dairesel olmayan bölümlerden oluşuyorsa, bu bölümler yaklaşık olarak doğrusal kabul edilebilecek çok daha küçük bölümlere ayrılır. Her küçük bölüm için koordinat hesaplama işlemlerinin yapılması gerekir; ki bu da, Şekil 4-25'de gösterilen basit bir iş parçası için bile çok büyük bir iş yapılması demektir.

APT'de bir çevresel işleme hareketi, bitiş noktası koordinatları esasına göre değil, verilen parça profili tarafından tanımlanır. Ayrıca, bir parçanın profili, APT'de tanımlanabilir geometrik cisimlerin herhangi birisi olabilir. Bu nedenle, çevresel işleme hareketinin tanımlanması iyice basitleştirilmiş olur.

5.4.1 Çevresel İşleme Hareketinin Üç Kontrol Yüzeyi: Tahrik (sürücü) Yüzeyi, Parça Yüzeyi, Kontrol Yüzeyi

Geometrik bakımdan; bir kesici takım hareketinin başlangıç ve bitiş noktaları, kesici takımın o anda içinde bulunduğu hareket yolu ile, sırasıyla bir önceki ve bir sonraki yollarının keşistikleri noktalarıdır. Bu noktalar NC işlemcisi tarafından kolaylıkla hesaplanabilir; çünkü her bir kesici takım yolu bir geometrik cisim olarak tanımlanabilir. Çevresel işleme hareketinin bir aşamasında, kesicinin uzayda yaptığı hareket iki şekilde karakterize edilir: Kesici takım eksenindeki hareket ve bu eksene dik olan düzlemdeki hareket. Örneğin üç eksenli bir freze tezgahı için; bunlar, kesicinin z eksenini boyunca ve X-Y düzlemine paralel bir düzlem içinde yaptığı hareketlerdir. Kesici hareketinin her bir aşaması için bilinen bir başlangıç



Şekil 5-9 Çevresel işleme hareketini kontrol eden üç yüzey.

noktası mevcuttur ve bu nokta, söz konusu hareketten bir önce gelen aşamanın bitiş noktası olup, bir önceki deyimle tanımlanır. Bu nedenle, bir kesici hareketi; kesicinin hareketini kesici eksenine boyunca sınırlayan bir yüzey, kesici eksenine dik bir düzlemde sınırlayan bir yüzey ve hareketin bitiş noktasını tanımlayan bir yüzey vasıtasıyla tanımlanabilir.

APT dilinde sırasıyla parça yüzeyi, tahrik yüzeyi ve kontrol yüzeyi olarak adlandırılan bu üç yüzey şu şekilde tanımlanırlar (Şekil 5-9):

- Tahrik yüzeyi: Kesme işlemi süresince kesiciyle sürekli temas halinde bulunan ve kesici hareketini, kesici eksenine dik düzlemde yönlendiren yüzeydir.
- Parça yüzeyi: Kesme işlemi süresince kesiciyle sürekli temasta bulunan ve kesici eksenine boyunca kesicinin hareketini ya da kesme derinliğini kontrol eden yüzeydir.
- Kontrol yüzeyi: Belirtilen çevresel işleme hareketinin bitiş noktasını ya da bir sonraki hareketin başlangıç noktasını belirleyen yüzeydir.

Herbir çevresel işleme hareketi deyimi için, bu üç yüzeyin doğru biçimde tanımlanması şarttır. Dolayısıyla bu üç yüzeyin birbirlerinden ayırdedilmesi, APT'de bir çevresel işleme hareketinin tanımlanması için en önemli husustur. Aşağıdaki örneklerde; tahrik, parça ve kontrol yüzeyleri kavramı açıklanmaktadır.

Şekil 5-10(a)'da, tüm sınırlayıcı yüzeyler düzlemdir. Kutu şeklindeki parçayı frezede işlemek için gerekli kesici hareketi oklarla gösterilmiştir. Hareketin her bir aşamasına ait olan bu üç yüzey aşağıdaki gibidir:

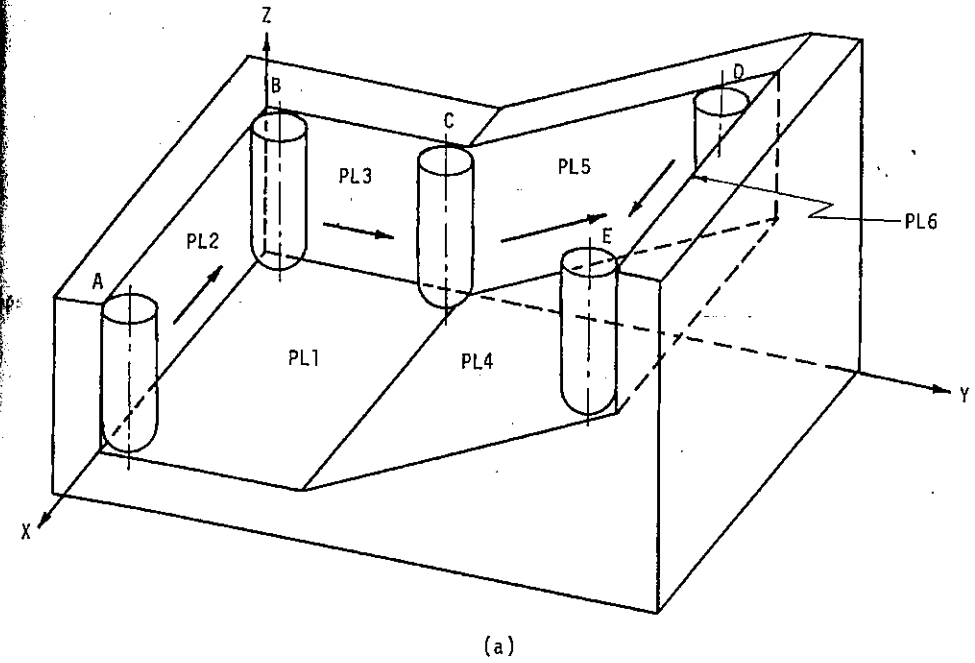
HAREKET	PARÇA YÜZEYİ	SÜRME YÜZEYİ	KONTROL YÜZEYİ
---------	--------------	--------------	----------------

A'dan B'ye	PL1	PL2	PL3
B'den C'ye	PL1	PL3	PL4
C'den D'ye	PL4	PL5	PL6
D'den E'ye	PL4	PL6	PL7 (şekilde gösterilmemiştir)

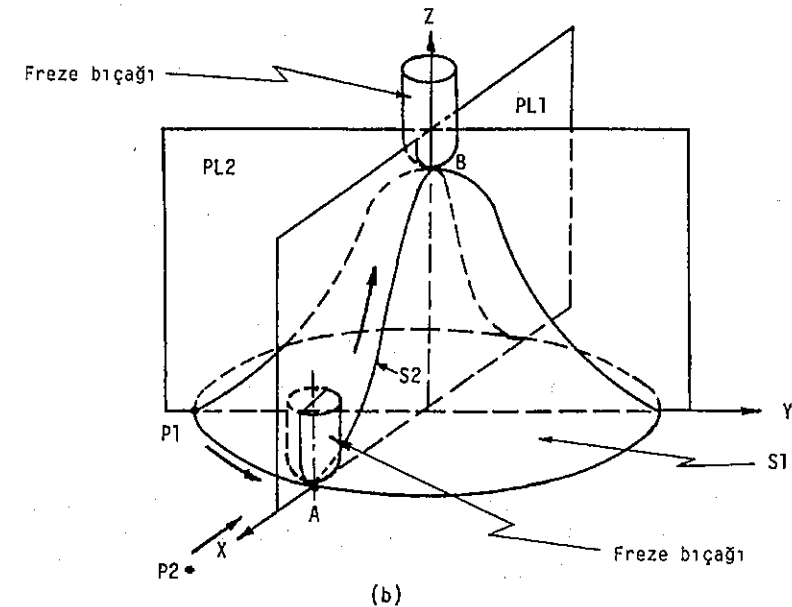
C'den D'ye hareket için parça yüzeyi PL4'tür; çünkü Z yönündeki doğrultusundaki kesici takım hareketini belirlemektedir.

Şekil 5-10(b)'de verilen parça, çan şeklindedir. A'dan B'ye Z doğrultusunda kesici takım hareketi, S1 yüzeyi ile PL1 düzleminin arakesiti olan S2 tarafından kontrol edilmekte; kesici takım eksenine dik düzlemdeki kesici takım hareketinin kontrolü ise PL1 düzlemi tarafından sağlanmaktadır. Dolayısıyla S2, PL1 ve PL2 yüzeyleri; sırasıyla parça, tahrik ve kontrol yüzeyleridir.

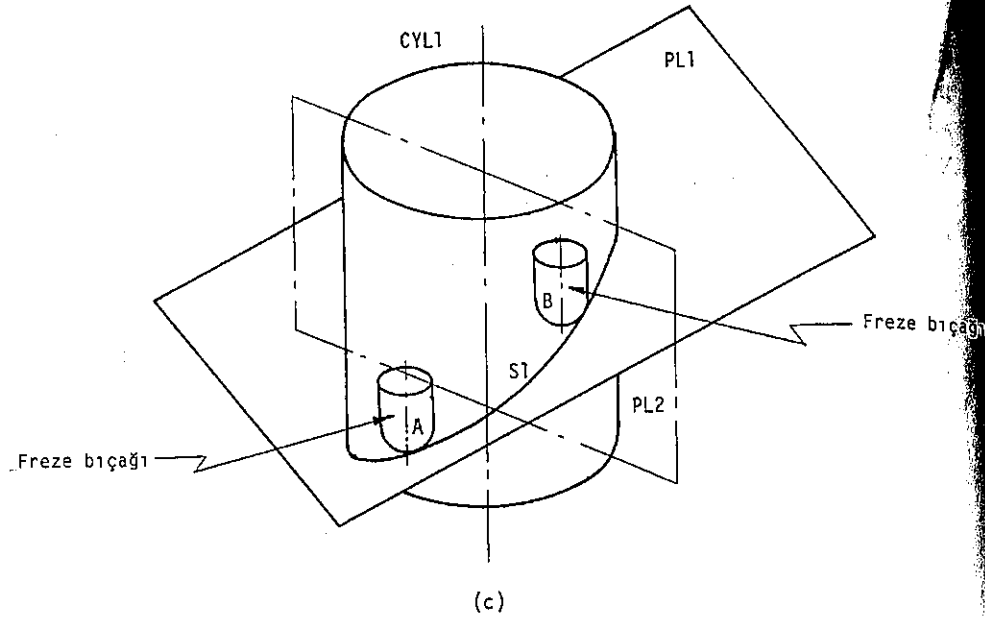
Şekil 5-10(c)'de, kesici takımın, PL1 düzlemiyle CYL1 silindirinin arakesi-



Şekil 5-10(a) Cep açma işlemi için freze bıçağı hareketi.



Şekil 5-10(b) Çan şeklinde bir parça için freze bıçağı hareketi ve kontrol yüzeyleri.



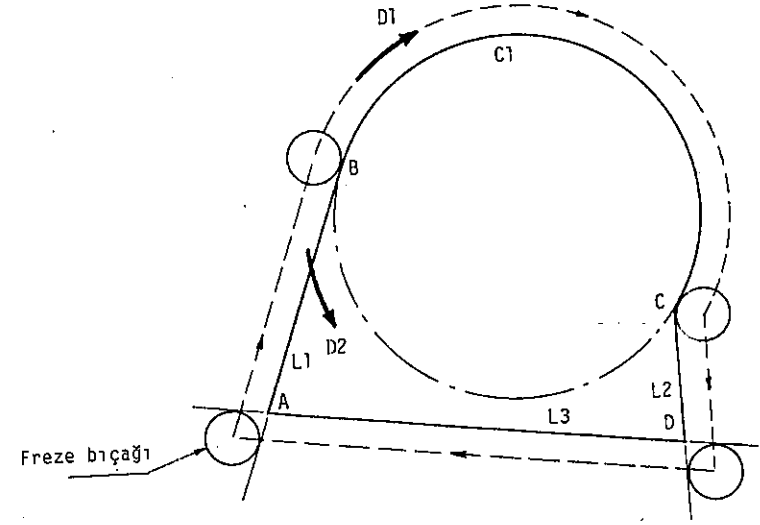
Şekil 5-10(c) PL1 düzlemi ile CYL1 silindirin arakesiti tarafından şekillendirilmiş bir helis boyunca çevresel işleme hareketi.

tinin bir parçası olan bir helis boyunca A'dan B'ye hareket ettirilmesi gerekmektedir. Kesici takım eksenine dik düzlemde kesici takım hareketi CYL1 silindiri tarafından, aksenal hareketi ise PL1 düzlemi tarafından, kontrol edilmektedir. Buna ek olarak, B konumu, PL2 düzlemi ile CYL1 silindirin arakesiti üzerinde yer almaktadır. Dolayısıyla parça, tahrik ve kontrol yüzeyleri sırasıyla PL1, CYL1 ve PL2'dir. CYL1'i sürücü yüzey olarak alıp S1 eğrisini parça yüzeyi olarak kullanmak, ya da S1 eğrisini sürücü yüzey, PL1 düzlemini de parça yüzeyi olarak kullanmak da mümkündür. Bu nedenle; üç kontrol yüzeyi; bir yüzey, bir düzlem, bir uzay eğrisi veya bir doğru olabilir.

5.4.2 Çevresel İşleme Hareketini Tanımlayan Deyim

APT'de, yukarıda belirtilen üç yüzey, çevresel işleme hareketini tanımlamak üzere kılavuz yüzeyler olarak kullanılır. Kural olarak, üç yüzeyi belirlemek için verilen bir parçanın dizayn profilini kullanırız. Bu APT deyimleriyle tanımlanan bir kesici takım konumu, kesici ucu merkezinin konumudur. Her kesici takımın belli boyutları olduğu için, takımı bu üç kızaklama yüzeyi boyunca hareket ettirmek tasarımdan farklı bir profil yaratacaktır. Kesici, freze bıçağı yarıçapını telafi etmek için, kesiciyi tasarım profilinin sağ ya da sol tarafında konumlamak gereklidir.

Bir APT deyimini tarafından tanımlanmış bir çevresel işleme hareketi, her



Şekil 5-11 Şekillendirilmiş bir parça için çevresel işleme hareketi. PL1 parça yüzeyi (burada gösterilmemiştir) kesici takım eksenine dik durumdadır.

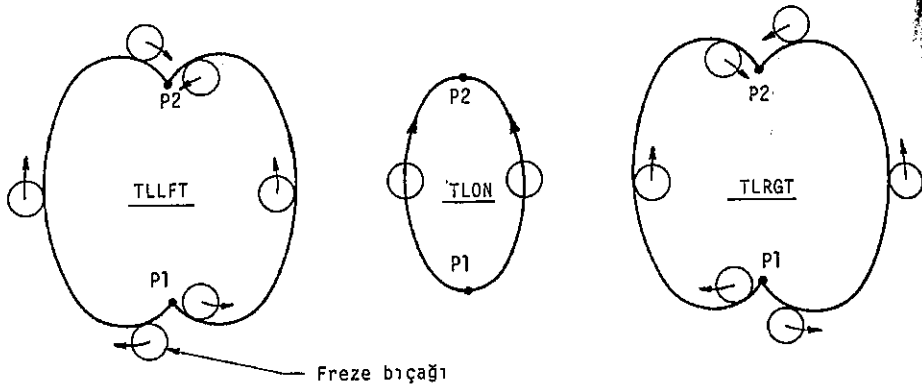
zaman bir önceki harekete göre verilir. NC programlamalarında kontrol elemanı hareket yönünü tanımlamak amacıyla karıştırılmaz; çünkü nihai varış yeri, koordinatları ve, aynı zamanda hareket yönünü de gösteren (G02 ya da G03 gibi) bir G koduyla tanımlanır. Buna karşın APT düzeninde bir hareket üç yüzey tarafından tanımlanır. Bunlardan bir tanesi doğru olmasına rağmen, çoğu durumda kesici takımın izleyebileceği iki muhtemel doğrultu vardır. Örneğin Şekil 5-11'de B konumuna varan kesici takım, bir sonraki hareket için, freze bıçağı hareketinin sürme yüzeyi olarak C1 çemberiyle, kontrol yüzeyi olarak da L2 düğrusuyla tanımlandığı D1 ya da D2 yönlerinden birini izleyebilir. Dolayısıyla kesici takım hareketinin yönünü önceki hareketi dikkate alarak tanımlamak gereklidir.

Son olarak; bir konturlama hareketi deyiminde, kesici takım tarafı merkezinin konumunu parça yüzeyine göre belirlemek gereklidir. Bu, özellikle üç boyutlu işleme için çok önemlidir.

Bir çevresel işleme hareketi deyiminin genel formatı aşağıdaki gibidir:

$$\left[\begin{array}{l} \{TLON\} \\ \{TLLFT\} \\ \{TLRGT\} \end{array} \right] \left[\begin{array}{l} \{L,TL\} \\ \{OFFS\} \\ \{,TLONPS\} \end{array} \right] \left[\begin{array}{l} \{GOLFT\} \\ \{GORGT\} \\ \{GOFWD\} \\ \{GOBACK\} \\ \{GOUNP\} \\ \{GODOWN\} \end{array} \right] /S1 \left[\begin{array}{l} \{L,TQ\} \\ \{ON\} \\ \{,PAST\} \\ \{,TANTO\} \\ \{,PSTAN\} \end{array} \right] ,S2[A]$$

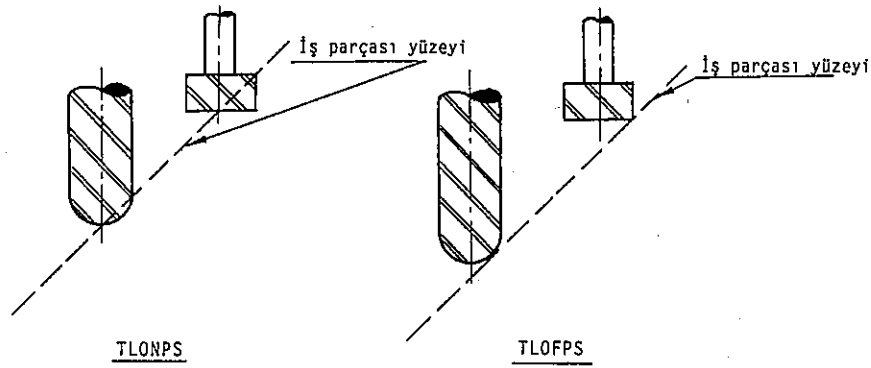
Burada; S1, S2 ve f sırasıyla tahrik yüzeyi, kontrol yüzeyi ve ilerleme hızı simgeleridir. Deyimde belirtilen dört kelime ise aşağıda açıklanmaktadır. İlk kelime (yani; TLON, TLRGT, ya da TLLFT) belirtilen tahrik yüzeyine ya da



Şekil 5-12 Belirtilen tahrik yüzeylerine göre kesici takım konumları. Oklar hareket yönünü göstermektedir. P: Başlangıç noktası; P2: bitiş noktası

tasarlanmış profile (Şekil 5-12) göre kesici takımın konumunu göstermektedir. TLON kelimesi; kesici takım tahrik yüzeyi boyunca hareket ederken, kesici uç merkezinin mutlaka tahrik yüzeyi üzerinde yerleştirilmiş olması gerektiğini gösterir. TLLFT ve TLRGT kelimeleri ise, kesici takımın sırasıyla tahrik yüzeyinin sağ ve sol taraflarında ve daima yüzeye teğet olması gerektiğini belirtir. Bu kelime, bir önceki deyimde belirtilenle aynı olduğu takdirde, atlanır.

Deyimdeki ikinci kelime (yani TLONPS ya da TLOFPS), parça yüzeyine göre kesici takım uç merkezinin bağıl konumunu göstermektedir (Şekil 5-13). Eğer TLONPS kelimesi belirtilmişse, kesici uç merkezi hareket sırasında her zaman parça yüzeyinin üzerindedir. TLOFPS kelimesi: (1) kesici takım profilinin iş parçası yüzeyine her zaman teğet ya da merkezlenmemiş durumda olduğunu, (2)

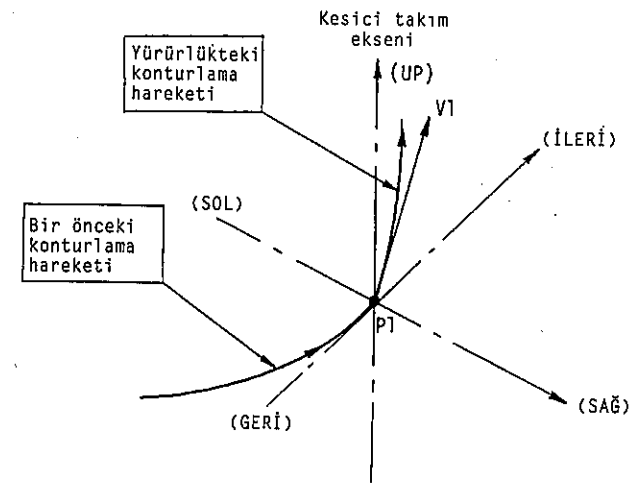


Şekil 5-13 İş parçası yüzeylerine göre TLONPS ve TLOFPS kelimeleriyle tanımlanmış kesici takım konumları.

Düz uçlu ve iş parçası yüzeyine dik durumdaki takımlar dışında, kesici takım ve iş parçası yüzeyinin sadece bir ortak noktası ve doğrusu olduğunu gösterir. Eğer bu kelimelerden hiç birisi belirtilmemişse, bu aynı zamanda kesici takımın hatalı konumlandığını da ifade eder.

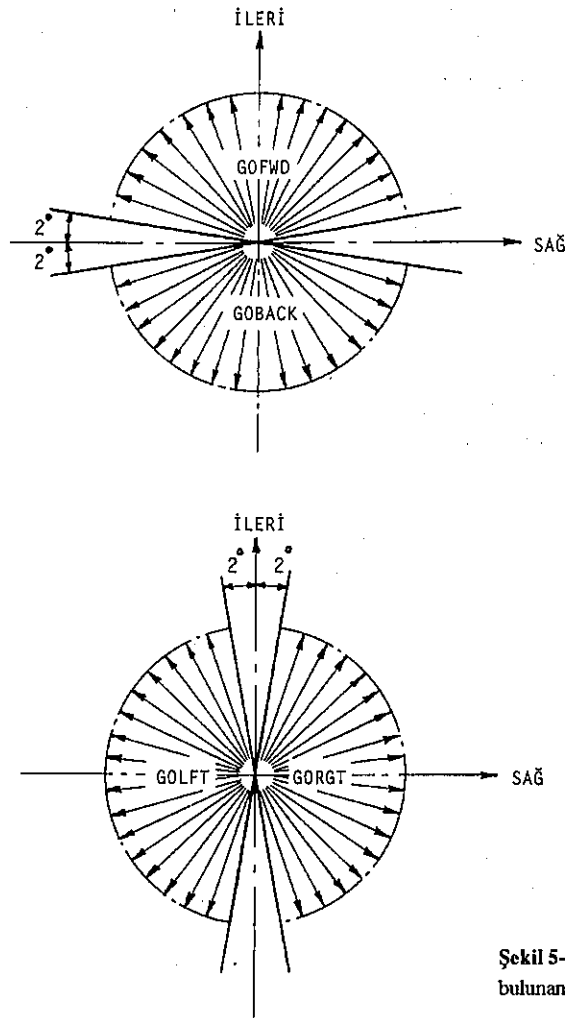
Üçüncü kelime; GOLFT, ..., ya da GODOWN; bir önceki hareket yönüne göre bu deyim tarafından tanımlanan kesici takım hareketinin yönünü belirtir. GOLFT ve GORGT kelimeleri, bir önceki hareketin bitiş noktasına geldiğinde kesici takımın sırasıyla sola ve sağa dönmesi gerektiğini ifade eder. GOFWD, GOBACK, GOUP ve GODOWN kelimeleri ise, bir önceki hareketin bitiş noktasına vardığı zaman kesici takımın sırasıyla ileri, geri, yukarı ve aşağı gitmesi gerektiğini ifade eder.

Bir önceki hareketin bitiş noktasında (ki, bu nokta aynı zamanda yeni hareketin başlangıç noktasıdır) hareket yönünün değişmesi şöyle belirlenebilir: Kesici takım eksen vektörünü ucundan en üst noktasına kadar ve bir önceki hareketin hem sürme hem de parça yüzeylerine teğet olan vektörü çiziniz; böylece Şekil 5-14'te gösterilen UP (yukarı hareket) ve FORWARD (ileri hareket) yönlerini belirlemiş olursunuz. RIGHT (sağ) yön, FORWARD (ileri) ve UP (yukarı) vektörlerinin vektörel çarpımıyla belirlenir. Daha sonra, halen geçerli deyimden hem sürme hem de parça yüzeylerine teğet olan V1 vektörünü çiziniz. Genelde, hareket yönü aynı anda üç yönde değişebilir. Fakat biz sadece deyimde belirtilen tek bir yöndeki değişimi belirleyebildiğimiz için, önemli ölçüde değişikliğin meydana geldiği yönü bulmamız gerekir. Böylece, kesici takım kendi eksenini (Z) doğrultusunda konumunu değiştirmedikleri, ya da çok az değiştirdikleri takdirde, Z eksenine dik bir düzlem üzerine V1 vektörünün izdüşümünü alabiliriz. GOUP ve GODOWN seçenekleri



Şekil 5-14 Bir önceki çevresel işleme hareketinin bitiş noktasındaki hareket yönünün referans eksenleri.

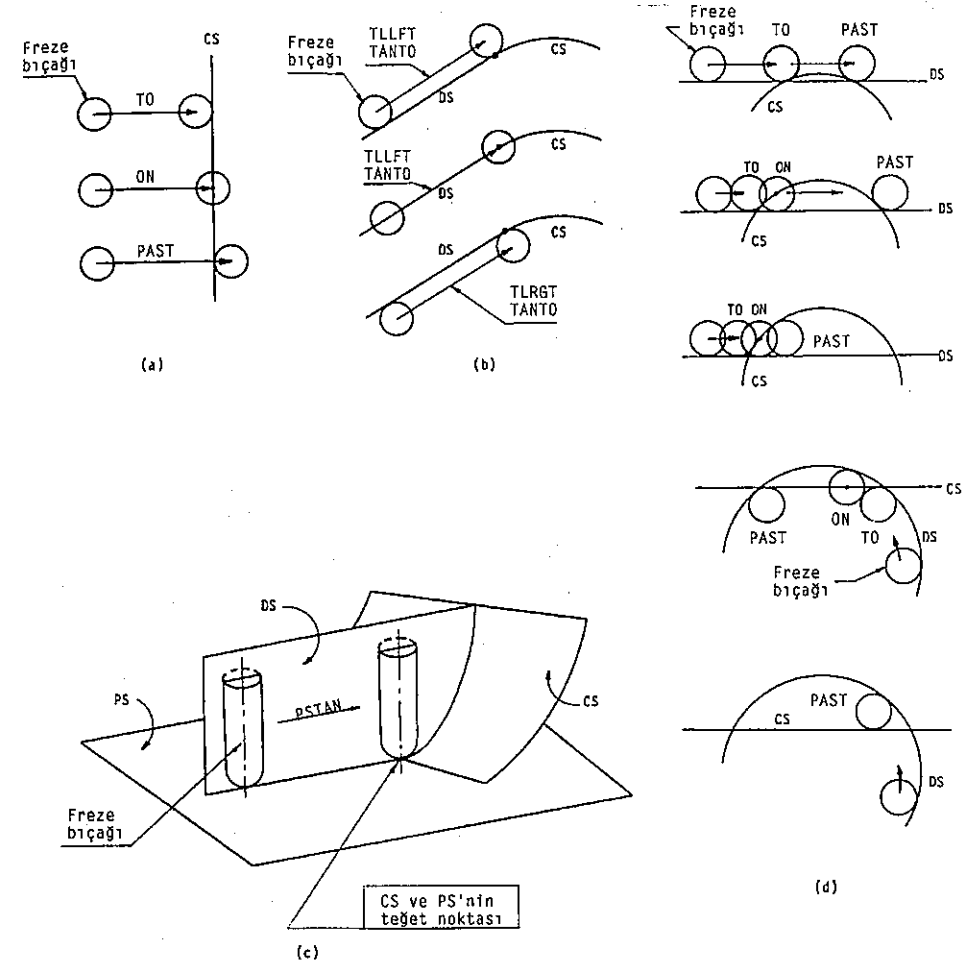
bu durumda dışarıda bırakılır. Geriye kalan GOFWD, GOBACK, GOLFT, veya GORGT kelimelerinden bir tanesi seçilir. Bu dört kelimedenden her birinin Şekil 5-15'ten de görüleceği gibi 176 derecelik bir açı belirleyen genel bir doğrultu anlamı gösterildiğine dikkat edilmelidir. Dahası, GOFWD ve GOBACK tarafından belirlenen doğrultular, GOLFT ve GORGT tarafından belirlenen doğrultularla çakışır. Dolayısıyla; V1 hareket vektörü ile FORWARD (ileri hareket) eksenine ya da V1 vektörü ile BACKWORD (geri hareket) eksenindeki açı 45 dereceden büyük olduğunda, GOFWD ve GOBACK kelimelerinin kullanılmaması tavsiye edilir. Bunların yerine GOLFT veya GORGT'un kullanılması tavsiye edilir. GOUP veya GODOWN kelimesi, kesici takım hareket yönündeki değişikliğin esas itiba-



Şekil 5-15 Çeşitli yön kelimeleri kapsamında bulunan yön aralıkları.

riyle düşey doğrultuda (kesici takım eksenine yönünde) olması halinde seçilir. Yukarıda belirtilen deyimdeki dördüncü kelime, hareket bitişinde kesici takım ile kontrol yüzeyi arasındaki konumsal ilişkiyi gösterir (Şekil 516). Seçilebilecek sözcükler şunlardır:

TO Kesici takımın, kontrol yüzeyine teğet olduğu noktada durduğunu gösterir. Bu kelime atlanabilir (Şekil 5-16[a ve d]).



Şekil 5-16 Farklı kelimelerle tanımlanmış çevresel işleme hareketi bitişinde CS ve PS'nin teğet noktası kesici takım konumları: (a) TO, ON ve PAST; (b) TANTO; ve (c) PSTAN. (d) Tahrik ve kontrol yüzeyleri arasındaki farklı konumsal ilişkilerden doğan muhtemel kesici takım konumları. DS: Tahrik yüzeyi; PS: Parça yüzeyi; CS: Kontrol yüzeyi.

ON Kesici takımın, merkezi kontrol yüzeyi üzerinde olduğu zaman durduğunu gösterir (Şekil 5-16[a ve d]).

PAST Kesici takımın kontrol yüzeyine geçmesini ve uzak taraftaki yüzeye teğet olduğu zaman durmasını sağlar (Şekil 5-16[a ve d]).

TANTO Tahrik ve kontrol yüzeylerinin birbirine teğet olduğunu gösterir. Kesici takım hareketinin bitiş noktası bu iki yüzeyin teğet noktasıdır (Şekil 5-16[b]).

PSTAN İş parçası ve kontrol yüzeylerinin birbirine teğet olduğunu ve kesici takım hareketi bitiş noktasının bu iki yüzeyin teğet noktası olduğunu gösterir (Şekil 5-16[c]).

Tahrik ve kontrol yüzeyleri arasındaki farklı konumsal ilişkilerden doğan kesici takım konumları, ve hareketleri tanımlamak için kullanılan konum değiştiricileri Şekil 5-16[d]'de gösterilmektedir.

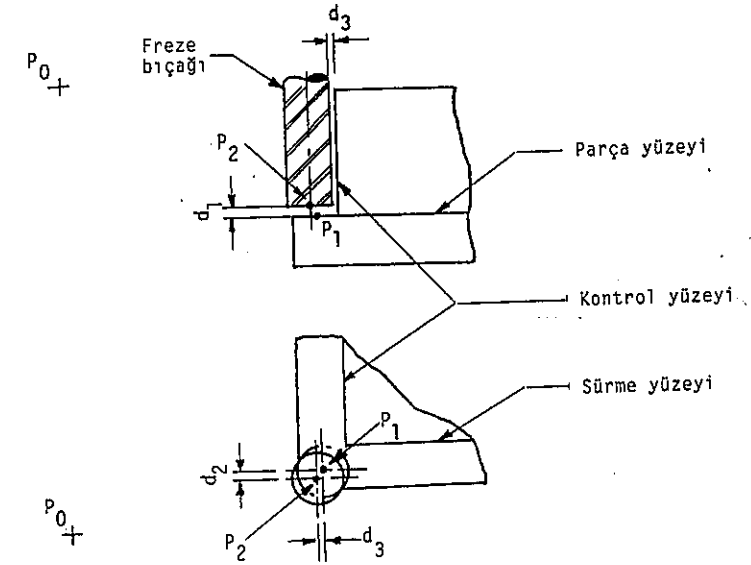
Aşağıdaki örnekler çevresel işleme hareketi deyiminin bazı uygulamalarını göstermektedir:

ŞEKİL	HAREKET	PARÇA YÜZEYİ	KONTURLAMA HAREKETİ DEYİMİ
5-10(a)	B'den C'ye	PL1	TLRGT,GORGT/PL3,TO,PL4
5-10(a)	C'den D'ye	PL4	TLRGT,GOUP/PL5,PL6
5-10(b)	(P1 →) A'dan B'ye	S2	TLON,GOLFT/PL1,ON,PL2
5-10(b)	(P2 →) A'dan B'ye	S2	TLON,GOFWD/PL1,ON,PL2
5-11	B'den C'ye	PL1	TLLFT,GOFWD/C1,L2
5-11	C'den D'ye	PL1	TLLFT,GOLFT/L2,PAST,L3

Bir çevresel işleme hareketi deyiminin, hareket sırasında kullanılacak parça yüzeyini gösteren hiçbir kelime içermediğine dikkat edilmelidir. Dolayısıyla; bu yüzey, ya bir çevresel işleme hareketini başlatan deyimde ya da parça yüzeyini tanımlayan ayrı bir deyimde tanımlanabilir.

5.5 BİR ÇEVRESEL İŞLEME HAREKETİNİ BAŞLATMA VE BİTİRME

Bölüm 5-4'te de belirtildiği gibi; çevresel işleme hareketi sırasında kesicinin tasarlanan profilden sapması, belirli bir tolerans içinde tutulmalıdır. Bu, kesici takımın, çevresel işleme hareketi başlangıcında yukarıda belirtilen şartın yerine getirildiği uygun bir noktada konumlanması gerektiğini ifade eder (Şekil 5-17). Çevresel işleme hareketi başlatılmadan önce, kesici takım tasarlanan profilden uzaktır. (1) Kesici takımı çevresel işleme hareketine başlamaya hazır konuma getirmek ve (2) Kullanılacak parça yüzeyini tanımlamak için bir ya da daha çok deyim gereklidir.



Şekil 5-17 Başlangıç hareketinin bitiş noktası ya da çevresel işleme hareketinin başlangıç konumu. P1: İstenilen bitiş konumu; P2: Gerçek bitiş konumu; P0: başlangıç hareketinin başlangıç konumu; d1,d2,d3: kesici takım konumunun sırasıyla parça, tahrik ve kontrol yüzeylerinden sapması.

Başlangıç hareketini tanımlamak için gerekli deyim, şu genel formata sahiptir:

$$GO / \left[\begin{array}{c} [TO,] \\ ON, \\ PAST, \end{array} \right] S1, \left[\begin{array}{c} [TO,] \\ ON, \\ PAST, \end{array} \right] S2, \left[\begin{array}{c} [TO,] \\ ON, \\ PAST, \\ TANTO, \end{array} \right], 53 [, f]$$

Burada;

GO, tanımlanan hareketin bir başlangıç hareketi olduğunu gösteren bir kelime; S1, S2, ve S3 sırasıyla tahrik, parça ve kontrol yüzeyleridir; TO, ON, PAST ve TANTO kelimeleri ise evvelce açıklanan aynı anlamı taşımaktadır (Herbir değiştirici, sadece kendisinden hemen sonra belirtilen yüzeye uygulanır).

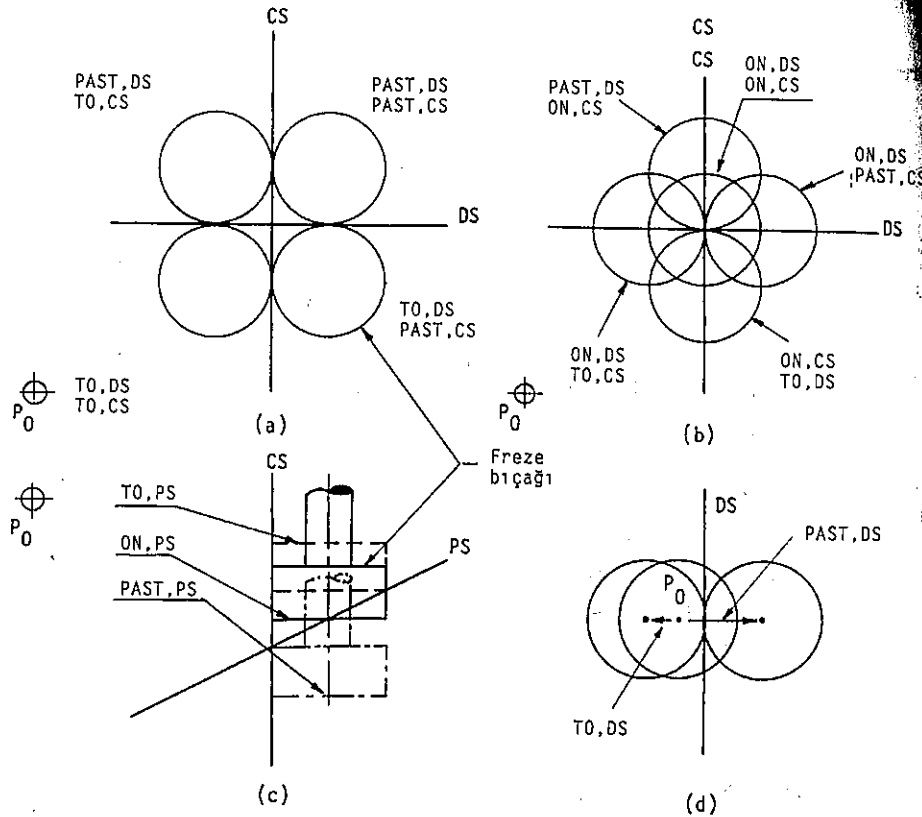
f; ilerleme hızı olup, evvelce tanımlananla aynıysa atlanabilir.

Başlangıç hareketinin bitişinde kesici takımın çeşitli konumları Şekil 5-18'de gösterilmektedir. Bu deyim sadece kesici takımı istenen konuma getirmek için kullanılır. NC işlemcisi bu hareketin yolunu hesaplamaz ya da dikkate almaz. Dolayısıyla; bu deyimle belirtilen hareket, üç kontrol yüzeyi belirtilmesine karşın, bir çevresel işleme hareketi değildir. Bitiş konumu ise; belirtilen yüzeyler,

değiştiriciler ve tolerans esas alınarak hesaplanır. Başlangıç hareketi esnasında kesicinin izleyeceği yol, GOTO deyiimiyle aynı biçimde, belirtilen hareket hızıyla (yani, bir RAPID ya da belirtilen ilerleme hızı) belirlenir. Bu deyimde tanımlanan parça yüzeyi, başka bir deyim tarafından değiştirilene kadar aşağıdaki çevresel işleme hareketinde etkili olacaktır.

Bazı durumlarda, kesici takımın tahrik yüzeyinin üzerinde belirli bir noktada konumlanmasına gerek yoktur. Böyle bir durumda, iki yüzeyli başlangıç hareketine ait şu deyim kullanılabilir:

$$GO / \begin{Bmatrix} [TO, I] \\ [ON, J] \\ [PAST, K] \end{Bmatrix} S1, \begin{Bmatrix} [TO, I] \\ [ON, J] \\ [PAST, K] \end{Bmatrix} S2$$

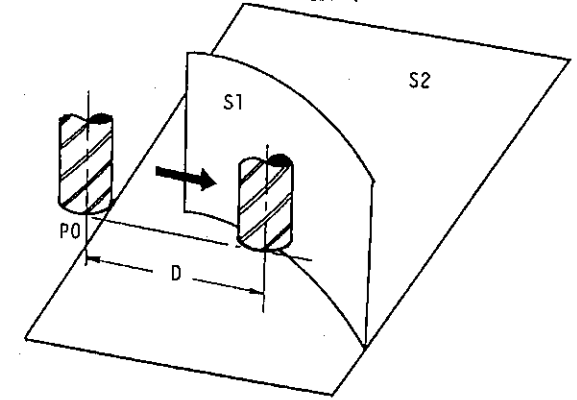


Şekil 5-18 Başlangıç hareketi sonunda kesici takımın çeşitli konumları. Başlangıç noktası PO'dır. (d) Bir kontrol yüzeyi kesici takımla kesiştiğinde başlangıç hareketinin bitiş konumlarına dikkat ediniz. DS: Tahrik yüzeyi; PS: Parça yüzeyi; CS: Kontrol yüzeyi.

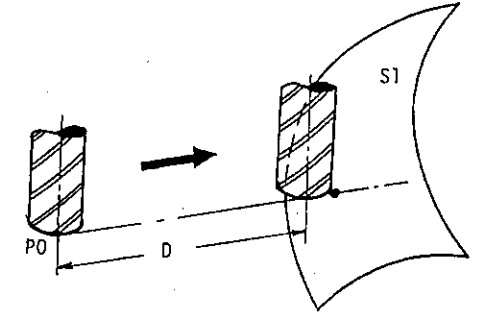
Burada; S1 ve S2 sırasıyla tahrik ve parça yüzeyleridir. Parça yüzeyi yeni bir parça yüzeyi tanımlanmaya kadar müteakip çevresel işleme hareketinde etkili olur. Genel olarak bu deyim tanımladığı bitiş konumu, kesici takımın, başlangıç noktasından tahrik ve parça yüzeylerinin arakesitine kadar mümkün olan en kısa mesafeyi katetmesini sağlayacak şekilde belirlenir (Şekil 5-19[a]).

Eğer bir kesici takımın bir sonraki çevresel işleme hareketi deyiimi içinde tahrik yüzeyi olarak kullanılacak tek bir yüzeye en kısa şekilde hareket etmesi gerekiyorsa, tek-yüzeyli yol verme hareketine ait şu deyim kullanılabilir:

$$GO / \begin{Bmatrix} [TO, I] \\ [ON, J] \\ [PAST, K] \end{Bmatrix} S1$$



(a) GO/S1, S2



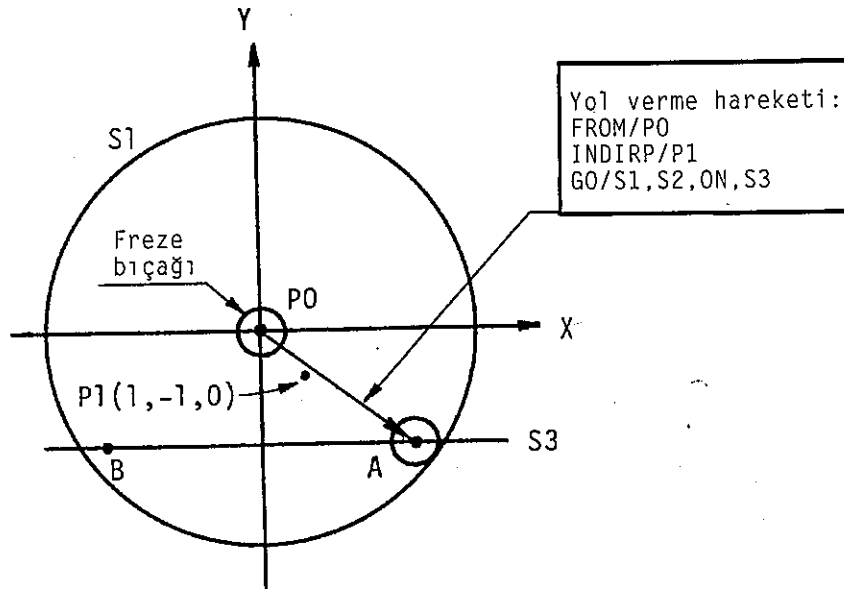
(b) NOPS;GO/S1

Şekil 5-19 Başlangıç hareketinin bitiş konumları. (a) İki yüzeyli başlangıç hareketi: D, başlangıç noktasından (S1) süme yüzeyi ile (S2) parça yüzeyinin ara kesitine kadar olan en kısa mesafedir. (b) Parça yüzeyi olmayan bir yüzeyli başlangıç hareketi: Bitiş konumu, tahrik yüzeyi ile bu yüzeyin PO başlangıç noktasından geçen normalinin kesişim noktası esas alınarak belirlenir.

Burada; S1'e APT programlama kılavuzunda tahrik yüzeyi adı verilmiştir, ve TO, ON ve PAST konum değiştiricileri yukarıda açıklananla aynı anlama sahiptir. Üç muhtemel konum vardır:

1. Parça yüzeyi bir önceki deyimde tanımlanmıştır. Bu durumda kesici takım hareketi iki-yüzeyle başlangıç hareketine benzerdir.
2. Parça yüzeyi tanımlanmıştır. Bu durumda, X-Y düzlemi parça düzlemi olarak kullanılır. Kesici takım hareketi yine iki-yüzeyle başlangıç hareketine benzer niteliktedir.
3. Tek-yüzeyle başlangıç hareketi deyiminden önce, başlangıç hareketi için parça yüzeyine gerek olmadığını belirtmek amacıyla bir NOPS deyim belirtilir (Şekil 5-19[b]). Böyle bir durumda SD işlemcisi, başlangıç noktasından geçen ve belirtilen S1 yüzeyine dik olan bir normal vektör yaratarak bitiş noktasını belirler.

Yukarıda sunulan açıklamadan da görüleceği gibi, S1 yüzeyi aslında bir tahrik yüzeyinden çok, başlangıç hareketi için kontrol yüzeyidir. Sürme yüzeyi terimi, burada, S1 çoğu kez bir sonraki çevresel işleme hareket deyiminde sürme yüzeyi olarak kullanıldığı için kullanılmaktadır. Bu özellikler, iki-yüzeyle başlangıç hareketi deyiminde belirtilen ilk yüzeye de uygulanabilir.



Şekil 5-20 İstenen bir çözümü göstermek için INDIRV ya da INDIRP deyiminin kullanılması. S2 parça yüzeyi (şekilde gösterilmemiştir), X-Y düzlemine paraleldir.

Bütün hareket deyimleri gibi, başlangıç hareketi deyimini de bir hareketi verilen bir başlangıç konumuna göre tanımlar. Bu konum, bir programın başlangıcında bir FROM deyimine ya da programın ortasında bir önceki hareket deyimine tanımlanabilir.

NC işlemcisi içinde bir başlangıç hareketinin bitiş noktasını belirleme işlemi, verilen koşulları sağlayan matematiksel çözümleri bulmaya bağlıdır. Çoğu durumda çözüm tek değildir: iki ya da daha fazla çözüm olabilir (Şekil 5-20). Dolayısıyla, NC işlemcisinin istenilen çözümü seçebilmesi için ek bilgi sağlanmalıdır.

Bu amaçla kullanılan deyimler, INDIRV (verilen bir vektör yönünde) ve INDIRP (verilen bir noktanın yönünde) olup; bunlara ait formatlar ise şöyledir:

$$\text{INDIRV} \left\{ \begin{matrix} V1 \\ i,j,k \end{matrix} \right\}$$

burada; V1 verilen bir vektörü; i, j ve k ise bu vektörün sırasıyla X, Y ve Z yönlerindeki bileşenlerini ifade eder:

$$\text{INDIRP} \left\{ \begin{matrix} P1 \\ x,y,z \end{matrix} \right\}$$

burada; P1 verilen bir noktayı simgeler; x, y ve z ise bu noktanın koordinatlarıdır. Bu deyimler, bir başlangıç hareketinden hemen sonraki hareket deyiminde belirtilen başlangıç hareketi bitiş noktasının, bir önceki deyim bitiş noktasından başlayan V1 yönünde veya bir önceki deyim bitiş noktasından itibaren belirtilen noktaya kadar olan doğrultuda olduğunu gösterir. Örneğin; eğer verilen nokta P1 (1,-1,0) ise (Şekil 5-20), bir sonraki programdaki INDIRV veya INDIRP deyimini, yol verme hareketinin A bitiş noktasının bulunduğu yönü gösterir:

```
.....
.....
FROM/P0
INDIRP/P1$$ VEYA INDIRV/1,-1,0
GO/S1,S2,ON,S3
.....
```

Bu programın tanımladığı başlangıç hareketi, P0 noktasından A noktasına kadardır. INDIRP ya da INDIRV deyiminde verilen nokta veya vektörün hassas değerde olmasının gerekli olmadığına dikkat edilmelidir; çünkü bu iki deyim, sadece bir çözümü diğerinden ayırtmak için kullanılan genel bir yönü tanımlar. Dolayısıyla, Şekil 5-20'de gösterilen durumda, eğer A istenilen çözümse, o zaman P1 noktası

dördüncü çeyrek dairede herhangi bir nokta olabilir.

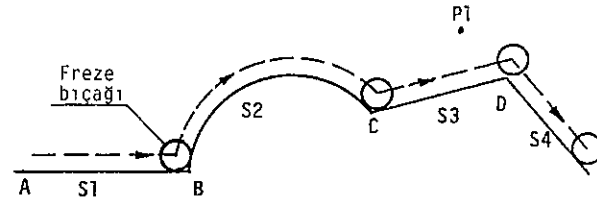
INDIRV ya da INDIRP deyiminin kullanılması başlangıç hareketiyle sınırlı değildir; bu deyimlerin her kisi de, çevresel işleme sırasında hareket yönündeki değişimi tanımlamak için kullanılabilir. Örneğin Şekil 5-21'de çevresel işleme hareketi A'dan B ve C üzerinden D'ye doğrudur. Programın bu hareket yolunu tanımlayan kısmı şöyledir:

.....
....(Kesici takım konumu B'dir)....

A1)TLLFT,GOLFT/S2,S3

A2)GOLFT/S3,PAST,S4

.....
.....



Şekil 5-21 Bir çevresel işleme.

Kesici takım C noktasından D noktasına hareket etmeye başladığı zaman C noktasındaki hareket yönünü belirtmek için, A1 ve A2'de gösterilen deyimlerin arasına bir INDIRP deyimini yerleştirilebilir. Program aşağıdaki gibi değiştirilebilir:

.....

....(Kesici takım konumu B'dir)....

A1)TLLFT,GOLFT/S2,S3

INDIRP/P1

A2)GOFWD/S3,PAST,S4

.....
.....

Yerleştirilen deyim, kesici takım C noktasına vardıktan sonra genel bir yön gösterir. Dolayısıyla A2'de gösterilen deyimdeki GOLFT kelimesi, GOFWD (ya da GORGT) olarak değiştirilmelidir. Bu, A2'de belirtilen çevresel işleme hareketi deyimindeki hareket yönü kelimesinin, kesici takım hareket yönünün değil, INDIRP deyimini tarafından belirtilen yönünün esas alınmasıyla seçilmesini gösterir.

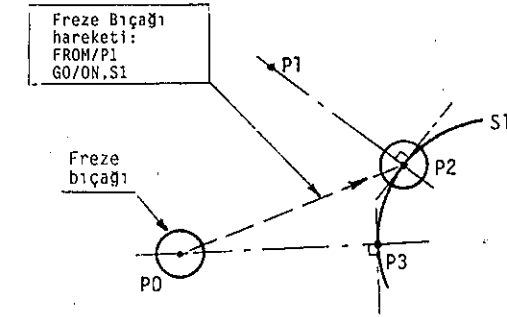
FROM deyimini bir başlangıç hareketinde kullanıldığında, bitiş noktasını hesaplamak için bir referans noktası oluşturur. Gerek INDIRV veya INDIRP deyimini tarafından gösterilen yön ve gerekse başlangıç hareketi deyimiyile tanımlanan konum, FROM konumuna göre dikkate alınır. Bir FROM konumu, kesici takımın gerçek konumu ile aynı veya farklı olabilir. Eğer farklı ise, başlangıç hareketinin bitiş noktasını belirlemek için (kesici gerçek konumu değil), FROM konumu kullanılır.

Buna göre, örneğin (Şekil 5-22) gerçek kesici takım konumunun P0 olduğunu ve çevresel işleme hareketini başlatmak için aşağıdaki deyimlerin kullanıldığını varsayalım:

.....
FROM/P1

GO/ON,S1

.....



Şekil 5-22 P1 FROM konumu P0 gerçek kesici takım konumundan farklı olduğu zaman, başlangıç hareketinin bitiş konumu.

Başlangıç hareketinin bitiş konumu, P3 yerine P2 olup; meydana gelen hareket P0'dan P2'ye doğrudur. Eğer FROM konumu, bir önceki hareket deyimini tarafından tanımlanmış gerçek kesici takım konumuyla aynı ise, FROM deyimini atlanabilir. Bir çevresel işleme hareketi sadece GOTO deyimini belirterek sona erdirilebilir. Kesici takım, bir çevresel işleme hareketinin bitiş konumundan, GOTO deyimiyile belirtilen konuma hareket edecektir. Kesici takımın hareket yolu yine; çevresel işleme hareketinin bitiş konumundan, GOTO deyimiyile tanımlanmış konuma doğru düz bir çizgi olmak zorunda değildir; bu belirtilen ilerleme hızına bağlıdır.

5.6 BİR ÇEVRESEL İŞLEME HAREKETİ İÇİN PARÇA YÜZEYİNİN TANIMLANMASI

Herbir çevresel işleme hareketi deyimi için bir parça yüzeyi gereklidir; çünkü bu, bir konturlama hareketini tanımlayan üç kontrol yüzeyinden biridir. Parça yüzeyi bir kez tanımlandıktan sonra, değiştirilene kadar etkili olur. Parça yüzeyi, iki-yüzeyli ya da üç-yüzeyli başlangıç hareketi deyimi içinde tanımlanabilir. Eğer bir-yüzeyli yol verme hareketi deyimi, kendisinden önceki bir NOPS deyimiyle belirtilmişse; parça yüzeyi tanımlanmaz. Sonuç olarak, bir sonraki çevresel işleme hareketi yapılamaz; çünkü NC işlemcisi Z yönündeki kesici takım hareketini belirleyemez.

İki-yüzeyli ve üç-yüzeyli başlangıç hareketi deyimlerinin yanısıra, APT'de bir parça yüzeyini tanımlamak için iki deyim sağlanmış olup; bunlara ait formatlar şöyledir:

PSIS/PL1

Burada; PL1 bu deyim tarafından bir sonraki konturlama hareketi için parça yüzeyi olarak tanımlanmış, verilen bir düzlemi belirten bir semboldür.

AUTOPS

ise; X-Y düzlemine paralel olan, ve bir sonraki çevresel işleme hareketi için parça yüzeyi olarak, mevcut kesici takım konumuyla aynı z koordinatına sahip düzlemi tanımlar.

Bir çevresel işleme hareketi sırasında kesici takım, daima bu üç kontrol yüzeyine göre belirlenmiş tolerans aralığı içinde konumlanır. Dolayısıyla, bir çevresel işleme hareketinde parça yüzeyi değiştirilecekse, yeni parça yüzeyi olarak kullanılacak yüzeye göre kesici takımın gerekli tolerans aralığı içinde olup olmadığı kontrol edilmelidir. Bu şartlar yerine getirilmişse, PL1'i bir sonraki çevresel işleme hareketinde parça yüzeyi olarak tanımlamak için

PSIS/PL1

komutu kullanılabilir. Aksi takdirde; yeni bir parça yüzeyi kullanılmadan önce, yeni bir çevresel işleme hareketi deyimi belirtilmelidir. Bölüm 5.8'de ilgili örnekler (Örnek 2 ve 3) verilmiştir.

5.7 KABA VE İNCE ÇEVRESEL HAREKETLERİNİN TANIMLANMASI

Bir iş parçasının, genellikle tek pasoda istenen ölçülere tam uygun olarak işlenemeyeceği ve yüzey hassasiyetinin elde edilemeyeceği, tezgahta işleme konusunda iyi bilinen bir husustur. Önce kaba ve yarı hassas yüzey işleme yapıldıktan sonra, hassas yüzey işleme elde edebilmek için bir veya daha çok sayıda paso vermek gereklidir. Kaba yüzey işleme esnasında, profil üzerinde hassas işleme için belli miktarda bir pay bırakılmalı ayrıca; kesici, istenen hassas işleme toleransına eşit bir masafe kadar profilden uzağa kaydırılmalıdır.

NC işlemcisi parça, tahrik ve kontrol yüzeyleri üzerine bırakılacak hassas işleme paylarını belirleyebilmemizi mümkün kılar ve böylece; çevresel işleme hareketinin istenen profilin tasarım esas alınarak tanımlanabilmesini sağlar. İşleme paylarını belirlemek için kullanılan deyim şöyledir:

THICK/a1[,a2[,a3]]

burada; $a1$, $a2$ ve $a3$ sırasıyla parça, tahrik ve kontrol yüzeylerindeki işleme paylarıdır. En son belirtilen işleme payı geri kalan tüm yüzeylere de uygulanır. Buna göre, eğer $a2$ ve $a3$ atlanmış ise, $a1$ payı tahrik, parça ve kontrol yüzeyleri için geçerli olur. Genellikle tahrik ve kontrol yüzeyleri için seçilmiş paylar aynı değere sahip olup; bu sayede, kesici takımın konumu, her bir hareket aşamasının sonunda yeni sürme yüzeyine göre daima istenen tolerans aralığı içinde kalmış olur.

Örneğin Şekil 5-23'de gösterilen parça için yapılacak kaba yüzey işlemesine ait kesici hareket yolu aşağıdaki gibi tanımlanabilir:

```
.....
FROM/P0
THICK/a1,a2
GO/ON,PL1,S0,S1
INDRV/0,1,0
TLLFT,GOFWD/S1,PAST,S2
GORGT/S2,S3
GOLFT/S3,.....
.....
```

Bu programda, tahrik ve kontrol yüzeylerinin toleransları aynıdır. Eğer tüm yüzeylerde eşit tolerans gerekiyorsa, $a1 = a2 = a3$ olur. Bu durumda, söz konusu deyim şu şekilde sadeleştirilebilir:

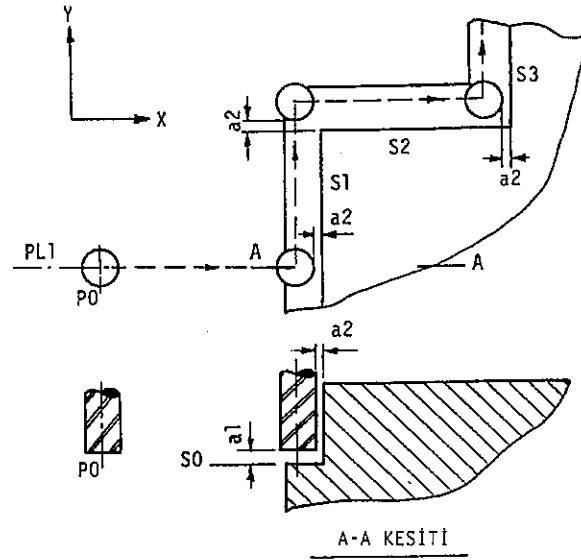
THICK/a1

Bir THICK deyimi, başka bir THICK deyimi tarafından değiştirilene kadar ken-

dinden sonraki bütün çevresel işleme hareketi deyimleri için etkilidir. Dolayısıyla, son işleme pasosu, THICK deyimindeki parametreleri sıfır değerine değiştirmek suretiyle tanımlanabilir.

Bir yol verme ya da çevresel işleme hareketinde bir kesici takım bir yüzey üzerinde konumlamak gerektiğinde (yani; bir ON değiştiricisi belirtildiğinde), bu yüzey için belirtilen işleme payı geçersiz sayılır. Örneğin, aşağıdaki programda, belirlenen işleme payı S2 parça yüzeyine uygulanmayacaktır:

```
.....
.....
THICK/0.02
GO/S1,ON,S2,PAST,S3
TLLFT,GORGT/S3:...
```



Şekil 5-23 (a_1, a_2, a_3) işleme paylarını belirten THICK deyiminin kullanılması.

5.8 ÇEVRESEL İŞLEME HAREKETİNİN PROGRAMLANMASI

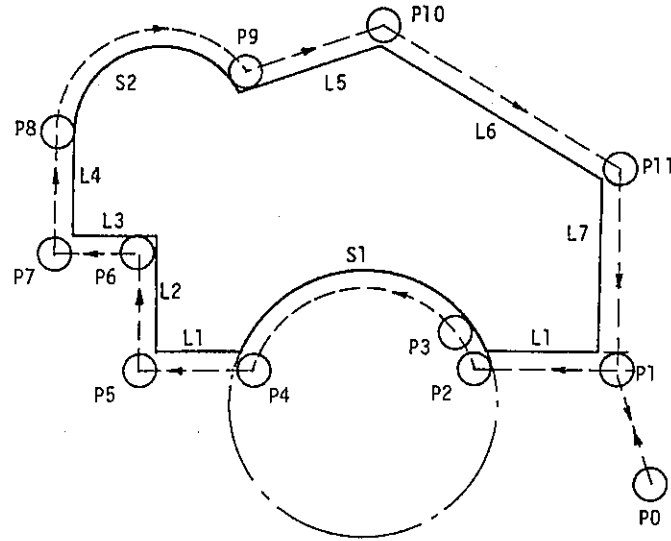
Yukarıda belirtilen açıklamalar esas alınarak, bir çevresel işleme hareketini tanımlamak için gerekli işlemler, sırasıyla aşağıdaki gibi özetlenebilir:

1. Başlangıç hareketinin tanımlanması
 - a. Başlangıç hareketini hesaplamak için NC işlemcisine ait referans noktasını tanımlayınız. Eğer bu konum mevcut kesici takım konumuyla aynı değilse, tanımlanması için bir FROM deyiminin kullanılması gerekir; aksi takdirde hiçbir deyim ihtiyacı yoktur.
 - b. Çeşitli yüzeyler için gereken işleme payını (veya paylarını) belirleyiniz.
 - c. Eğer gerekiyorsa, başlangıç hareketini tanımlamak için, INDIRV veya INDIRP deyimini ile birlikte bir başlangıç hareketi deyimini kullanınız.
2. Konturlama hareketinin tanımlanması
 - a. Bir konturlama hareketi; parça yüzeyi tanımlanmış olduğu için, iki-yüzeyle ya da üç-yüzeyle başlangıç hareketi deyiminden hemen sonra belirtilebilir. Kendinden önce bir NOPS deyimini kullanılarak, bir yüzeyli başlangıç hareketi deyimini seçilmişse; parça yüzeyinin, çevresel işleme hareketi deyimini belirtilmeden önce tanımlanması şarttır.
 - b. Kesici takım, başlangıç hareketinden sonra üç kontrol yüzeyine göre gerekli tolerans aralığı içinde konumlanmış olduğu için; bir sonraki konturlama hareketi deyiminde, kesicinin tahrik yüzeyi ya da kontrol yüzeyi tahrik yüzeyi olarak kullanılabilir.
 - c. Eğer çevresel işleme hareketi sırasında hareket yönü NC işlemcisi tarafından güçlükle algılanabiliyorsa, INDIRV ya da INDIRP deyimini eklenebilir. Bu deyim eklendikten sonra; bir sonraki çevresel işleme hareketinin yönü, INDIRV ya da INDIRP deyiminin gösterdiği yöne göre belirlenir.
 - d. Bir çevresel işleme hareketi deyiminin tahrik yüzeyi, bir önceki çevresel işleme hareketi deyimindeki sürme yüzeyi ya da kontrol yüzeyi olabilir.
 - e. Eğer parça yüzeyi çevresel işleme hareketi sırasında değiştirilecekse, kesici takımın bu yüzeye göre gerekli tolerans aralığı içinde konumlandığından emin olunuz.
3. Çevresel işleme hareketinin sona erdirilmesi. Bir çevresel işleme hareketi; noktadan noktaya harekete izin verilmişse bir GOTO deyimiyile; ya da eğer kesici takımın, parça profilinden uzaklaşmak üzere belirli bir yol izlemesi gerekli ise, bir çevresel işleme hareketi deyimiyile sona erdirilir.

Aşağıda, çevresel işleme hareketi deyimlerinin kullanılışını gösteren bazı örnekler verilmiştir.

Örnek 1

Şekil 5-24'de verilen parça, X-Y düzlemindeki doğrulardan ve eğrilerden oluşan bir profil göstermektedir. Kesici hareket yolunun P2'den P4'e doğru olacak şekilde programlanmasına dikkat edilmelidir. Kaba işleme ve ince (hassas) işleme için APT programı aşağıdaki gibi görünebilir:

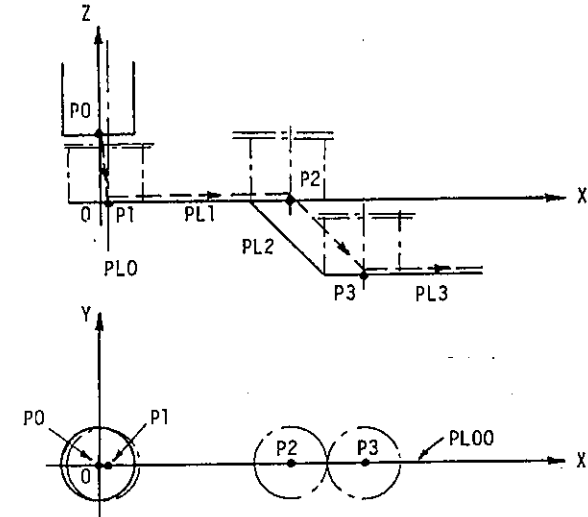


Şekil 5-24 Doğru ve eğrilerden oluşan bir parça profili. (Burada gözükmeyen) PS parça yüzeyi X-Y düzlemine paraleldir.

FROM/P0
THICK/0.01
GO/L7,PS,L1

1A) TLLFT, GOLFT/L1,PAST,S1
GORGT/S1,PAST,L1
GOFWD/S1,PAST,L1
GORGT/L1,PAST,L2
GORGT/L2,L3
GOLFT/L3,PAST,L4
GORGT/L4,TANTO,S2
GOFWD/S2,L5
GOLFT/L5,PAST,L6
GORGT/L6,PAST,L7
GORGT/L7,PAST,L1

SON NOKTA P1
SON NOKTA P2
SON NOKTA P3
SON NOKTA P4
SON NOKTA P5
SON NOKTA P6
SON NOKTA P7
SON NOKTA P8
SON NOKTA P9
SON NOKTA P10
SON NOKTA P11
SON NOKTA P1



IBM S/370 APT-AC N/C PROGRAM VERSION=1.3.0000

... BEGIN TRANSLATION PHASE... (SECTION 1) ...

```
ISN 00001 PARTNO TEST GO STATEMENT
ISN 00002 MACHIN/GN5CC,9,OPTION,2,0
ISN 00003 CLPRNT
ISN 00004 CUTTER/1.0
ISN 00005 FEDRAT/1.0
ISN 00006 PO=POINT/0,0,1
ISN 00007 PLO=PLANE/1,0,0,0,1
ISN 00008 PLOO=PLANE/0,1,0,0
ISN 00009 PL1=PLANE/0,0,1,0
ISN 00010 PL2=PLANE/1,0,1,2
ISN 00011 PL3=PLANE/0,0,1,-1
ISN 00012 FROM/P0
ISN 00013 GO/ON,PL00,PL1,ON,PL0
ISN 00014 INDIRV/1,0,0
ISN 00015 TLO,GOFWD/PL00,PAST,PL2
ISN 00016 PSIS/HL2
ISN 00017 GOFWD/PL00,TO,PL3
ISN 00018 PSIS/HL3
ISN 00019 GOFWD/PL00,TO,(PL4=PLANE/1,0,0,5)
ISN 00020 END
ISN 00021 FINI
```

NO DIAGNOSTICS ELICITED DURING TRANSLATION PHASE
17 N/C SOURCE RECORDS (SYSIN)

SECTION 1 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.1333
SECTION 2 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.0966

(b)

Şekil 5-25 (a) Parça yüzeyindeki değişimi içeren bir kesici hareket yolu. (b) (a)'da gösterilen kesici yolunu tanımlayan program.

2A)
THICK/O
GO/L7,PS,L1

(1A'dan 2A'ya kadar (dahil) olan deyimleri tekrarlayın)

GOTO/P0

Başlama ve bitiş noktaları P0'dır.

Örnek 2

Şekil 5-25(b)'de gösterilen program tarafından tanımlanan kesici hareket yolu, (Şekil 5-25[a]) parça yüzeyinde bir değişikliği de içermektedir. Kesici, P2 noktasına ulaşıldıktan sonra PL2 düzlemine göre gerekli tolerans aralığı içinde konumlanmış olduğu için, P2 noktasında bir yol verme hareketine gerek yoktur.

Herbir deyim için dahili sıra numarası (ISN), APT-AC NC işlemcisi tarafından işleme esnasında atanır ve bu numara, deyimın kendisine ait değildir.

Örnek 3

Şekil 5-26(b)'de gösterilen programda tanımlanan freze bıçağı hareket yolu (Şekil 5-26[a])'da C noktasında bir parça yüzeyi değişimini içermektedir. Parça yüzeyini değiştirmek için kullanılan ve kesiciyi aşağı hareket ettiren ISN 00018 etiketli deyim yerine şu iki deyim kullanılabilir:

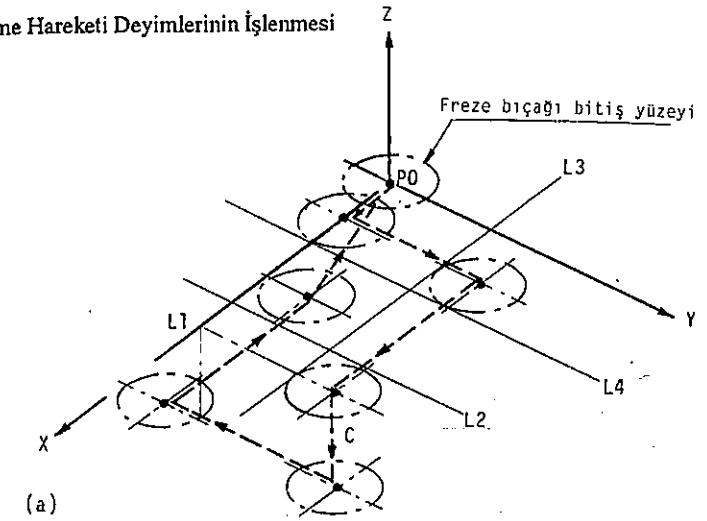
GODLTA/-1.0
AUTOPS

5.9 ÇEVRESEL İŞLEME HAREKETİ DEYİMLERİNİN İŞLENMESİ

Bu bölümde, bir dizi ardışık çevresel işleme deyiminin NC işlemcisi tarafından nasıl işlendiğini göreceğiz.

Matematiksel ifade ile, çevresel işleme hareketi deyimlerini işlemek, bir çevresel işleme hareketinin ardışık kesici konumlarını bulmak demektir. Böyle bir hesaplama için verilen durumlar şöyledir:

1. Tasarlanan ya da belirlenen profil
2. Tahrik, parça ve kontrol yüzeyleri için belirlenmiş işleme payları



IBM S/370 APT-AC N/C PROGRAM VERSION=1.3.0000

... BEGIN TRANSLATION PHASE... (SECTION 1) ...

```
ISN 00001 PARTNO TEST GO STATEMENT
ISN 00002 MACHIN/GN5CC,9,OPTION,2,0
ISN 00003 CLPRNT
ISN 00004 CUTTER/1.0
ISN 00005 FEDRAT/1.0
ISN 00006 P0=POINT/0,0,0
ISN 00007 L1=LINE/XAXIS
ISN 00008 L2=LINE/(2,0),(2,0)
ISN 00009 L3=LINE/(2,1),(0,1)
ISN 00010 L4=LINE/(1,1),(1,0)
ISN 00011 PL0=PLANE/0,0,1,0
ISN 00012 PL1=PLANE/0,0,1,-1
ISN 00013 FROM/P0
ISN 00014 GO/L4
ISN 00015 AUTOPS
ISN 00016 TLLFT,GOLFT/L4,PAST,L3
ISN 00017 GORGT/L3,PAST,L2
ISN 00018 GO/L2,PL1,L3
ISN 00019 GORGT/L2,PAST,L1
ISN 00020 GORGT/L1,TO,L4
ISN 00021 GOTO/P0
ISN 00022 END
ISN 00023 FINI
```

NO DIAGNOSTICS ELICITED DURING TRANSLATION PHASE
23 N/C SOURCE RECORDS (SYSIN)

SECTION 1 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.1333
SECTION 2 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.0966

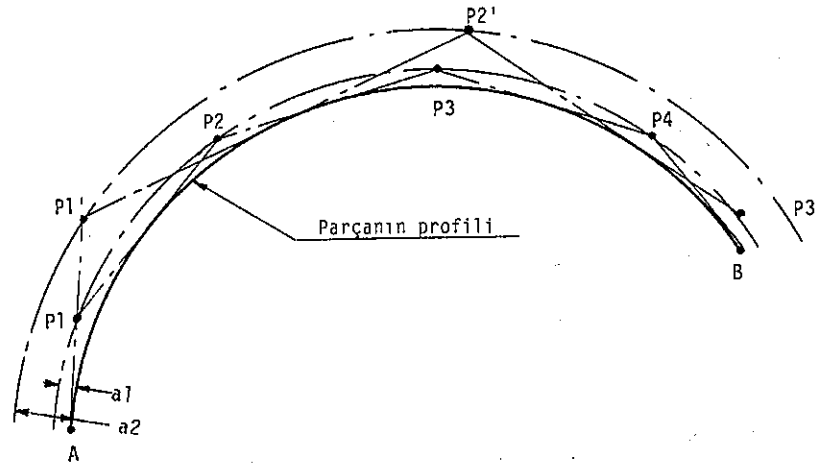
(b)

Şekil 5-26 (a) (b)'deki APT programıyla tanımlanmış bir kesici yolu. (b) Çevresel işleme hareketi esnasında parça yüzeyindeki değişimi gösteren bir APT programı.

3. Belirlenmiş tolerans(lar)*
4. Kesici takımın boyutları ve uzaydaki yönü*

Başlangıç hareketi için, kesici takımın bitiş konumu, verilen koşullar esas alınarak bulunur ve hesaplanmanın sonucu, başlangıç hareketi bitişinde kesici takım uç merkezinin konumunu tanımlayan bir koordinatlar takımından ibaret tek bir çıktıdır.

Bir çevresel işleme hareketi için, işlem sonucu, kesici yolunun (doğrusal ya da doğrusal olmayan) geometrik yerine bağlıdır. Eğer çevresel işleme hareketi doğrusalsa, sonuç başlangıç hareketi ya da noktadan noktaya hareket deyiminiyle aynıdır. Bu durumda, sadece bitiş konumu hesaplanır ve çıktısı elde edilir. Doğrusal olmayan bir çevresel işleme hareketinin işlenmesi ise çok daha karmaşıktır; çünkü hareket yolu, sadece bir dizi ardışık noktayla (ya da koordinat takımıyla) açıklanabilir. Örneğin Şekil 5-27'de gösterilen eğri biçimindeki yolun işlenmesi P1, P2, P3, ..., vb. noktaları ortaya çıkarır ve bu noktaların koordinatları CLDATA'daki çıktılardır. Aslında kesici takımın hareket yolu, bu hesaplanmış noktaları esas alan tasarlanmış profilin doğrusal enterpolasyonudur. Verilen bir profilin çevresel işleme hareketini açıklayan nokta sayısının belirtilen toleransla değiştiği bilinmektedir. Tolerans ne kadar dar olursa, kesici takım konumunu belirtilen tolerans aralığında muhafaza edebilmek için gerekli noktaların veya hareket aşamalarının (adımlarının) sayısı da o ölçüde artar. NC işlemcisinin tek bir çevresel işleme



Şekil 5-27 Bir çevresel işleme hareketinde ardışık kesici konumları. NC işlemcisi tarafından türetilen nokta sayısı, belirtilen toleransa (a1 ya da a2) bağlıdır.

* Tolerans ve kesici boyut özellikleri aynı deyimlerle belirlenir (Bölüm 6'da tartışıldı). Üç eksenli NC'li tezgahlar için kesici eksen daima Z yönündedir.

hareket deyimini için işleyebildiği en yüksek adım sayısı 400 olup; bu da genellikle yeterlidir. Bazı durumlarda, istenilen adım sayısı bu sınırı aşabilir. Böyle durumlarda yeni bir n limitini tanımlamak için

NUMPTS/n

deyimini kullanılabilir.

Her durumda, doğrusal olmayan bir çevresel işleme hareketinin işlenmesi sonucunda bir dizi nokta ortaya çıkar. Kesici hareketi bir noktadan diğerine doğrusal harekettir (çünkü genellikle RAPID dışında bir ilerleme hızı gerekir). Şekil 5-28(a ve b), dairesel bir hareket yolunu (bir çemberin dörtte biri) ve CLDATA çıktısını tanımlayan programı göstermektedir. Belirlenen 0.0005 inçlik tolerans için nokta sayısı 39'dur. Toleransı 0.0004 inç değerine düşürdüğümüz takdirde ise nokta sayısı 43'e yükselir (Şekil 5-28[c]). Listelenen programdaki CLPRNT deyimini SD işlemcisine, çoğu kez programlanmış bir kesici yolunu kontrol etmek için kullanılan CLDATA'yı basmasını söyler. CLDATA'ya ilişkin daha çok bilgi Bölüm 8 ve 11'de verilmektedir.

```

IBM S/370 APT-AC N/C PROGRAM VERSION=1.3.0000          DATE=08/05/86 TIME=15:04:24 0
... BEGIN TRANSLATION PHASE... (SECTION 1) ...
ISN 00001 PARTNO TEST GO STATEMENT
ISN 00002 MACHIN/GNSCC,9,OPTION,2,0
ISN 00003 CLPRNT
ISN 00004 CUTTER/0,0,5
ISN 00005 FEEDRAT/2,0
ISN 00006 TOLER/0,0005
ISN 00007 PO=POINT/0,0,0
ISN 00008 CI=CIRCLE/3,0,0,2
ISN 00009 LI=LINE/(3,0,0),(3,1,0)
ISN 00010 FROM/PO
ISN 00011 GO/CI
ISN 00012 AUTHORS
ISN 00013 TITLT,GOLEF/CI,ON,LI
ISN 00014 END
ISN 00015 FINI

```

NO DIAGNOSTICS ELICITED DURING TRANSLATION PHASE
15 N/C SOURCE RECORDS (SYSIN)

SECTION 1 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.0999
SECTION 2 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.0899
(a)

....SECTION 3....

```

ISN
0001 PARTNO/ TEST GO STATEMENT
0002 MACHIN/      GNSCC          9.00000000      OPTION      2.00000000
0.0
0004 CUTTER/      0.50000000
0005 FEEDRAT/     2.00000000
0006 CUTTOL/      0.00050000
0006 TINTOL/      0.0
0010 FROM/        PO
0.0
0011 GOTO/        C1
0.75000000      0.0      0.0
0013
0013 SURFACE      C1
0.00000000      0.0      0.0
0.0      1.00000000      2.00000000
0013 GOTO/        C1
0.74998067      0.04651086      0.0
0.75382479      0.13945312      0.0
0.76150646      0.23215713      0.0
0.77301256      0.32446450      0.0
0.78832344      0.41621752      0.0
0.80741292      0.50725945      0.0
0.83024840      0.59743473      0.0
0.85679087      0.68658931      0.0
0.88699497      0.77457086      0.0
0.92080911      0.86122906      0.0
0.95817551      0.94641588      0.0
0.99903034      1.02998575      0.0
1.04330378      1.11179592      0.0
1.09092021      1.19170659      0.0
1.14179827      1.26958126      0.0
1.19585104      1.34528687      0.0
1.25298616      1.41869408      0.0
1.31310603      1.48967748      0.0
1.37610793      1.55811578      0.0
1.44188422      1.62389207      0.0
1.51032252      1.68689397      0.0
1.58130592      1.74701384      0.0
1.65471313      1.80414896      0.0
1.73041874      1.85820173      0.0
1.80829341      1.90907979      0.0
1.88820408      1.95669622      0.0
1.97001425      2.00096966      0.0
2.05358412      2.04182449      0.0
2.13877094      2.07919089      0.0
2.22542914      2.11300503      0.0
2.31341069      2.14320913      0.0
2.40256527      2.16975160      0.0
2.49274055      2.19258708      0.0
2.58378248      2.21167656      0.0
2.67553550      2.22698744      0.0
2.76784287      2.23849354      0.0
2.86054688      2.24617521      0.0
2.95348914      2.25001933      0.0
3.00000000      2.25000000      0.0

```

0014 END
0015 ***** FINI *****
....END OF SECTION 3....

SECTION 3 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.0733

Şekil 5-28 (b) (a)'da gösterilen programın bilgisayar çıktısı (CLDATA). (devamı var)

Şekil 5-28 Dairesel çevre işleme hareketini tanımlayan bir APT programı. (devamı)

....SECTION 3....

```

ISN
0001 PARN/ TEST GO STATEMENT
0002 MACHIN/ GNSCC 9.00000000 OPTION 2.00000000
0.0
0004 CUTTER/ 0.50000000
0005 FEDRAT/ 2.00000000
0006 CUTTOL/ 0.00040000
0006 INTOL/ 0.0
0010 FROM/ PO
0.0 0.0 0.0
0011 GOTO/ C1
0.75000000 0.0 0.0
0013
0013 SURFACE C1 CIRCLE DS (IMP-TO)
3.00000000 0.0 0.0
0.0 0.0 1.00000000 2.00000000
0013 GOTO/ C1
0.74999346 0.04207993 0.0
0.75314030 0.12618093 0.0
0.75942957 0.21010546 0.0
0.76885248 0.29373614 0.0
0.78139586 0.37695600 0.0
0.79704215 0.45964866 0.0
0.81576948 0.54169845 0.0
0.83755166 0.62299063 0.0
0.86235821 0.70341151 0.0
0.89015445 0.78284859 0.0
0.92090150 0.86119080 0.0
0.95455636 0.93832855 0.0
0.99107195 1.01415396 0.0
1.03039721 1.08856098 0.0
1.07247714 1.16144556 0.0
1.11725288 1.23270575 0.0
1.16466182 1.30224188 0.0
1.21463764 1.36995672 0.0
1.26711046 1.43575555 0.0
1.32200687 1.49954634 0.0
1.37925011 1.56123988 0.0
1.43876012 1.62074989 0.0
1.50045366 1.67799313 0.0
1.56424445 1.73288954 0.0
1.63004328 1.78536236 0.0
1.69775812 1.83533818 0.0
1.76729425 1.88274712 0.0
1.83855444 1.92752286 0.0
1.91143902 1.96960279 0.0
1.98584604 2.00892805 0.0
2.06167145 2.04544364 0.0
2.13880920 2.07909850 0.0
2.21715141 2.10984555 0.0
2.29658849 2.13764179 0.0
2.37700937 2.16244834 0.0
2.45830155 2.18423052 0.0
2.54035134 2.20295785 0.0
2.62304400 2.21860414 0.0
2.70626386 2.23114752 0.0
2.78989454 2.24057043 0.0
2.87381907 2.24685970 0.0
2.95792007 2.25000654 0.0
3.00000000 2.25000000 0.0

```

0014 END

0015 ***** FINI *****

....END OF SECTION 3....

SECTION 3 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.0833

Şekil 5-28 (c) Tolerans 0.0004 inç değerine değiştirilmiş olarak, (a)'daki APT programı CLDATA'sı.

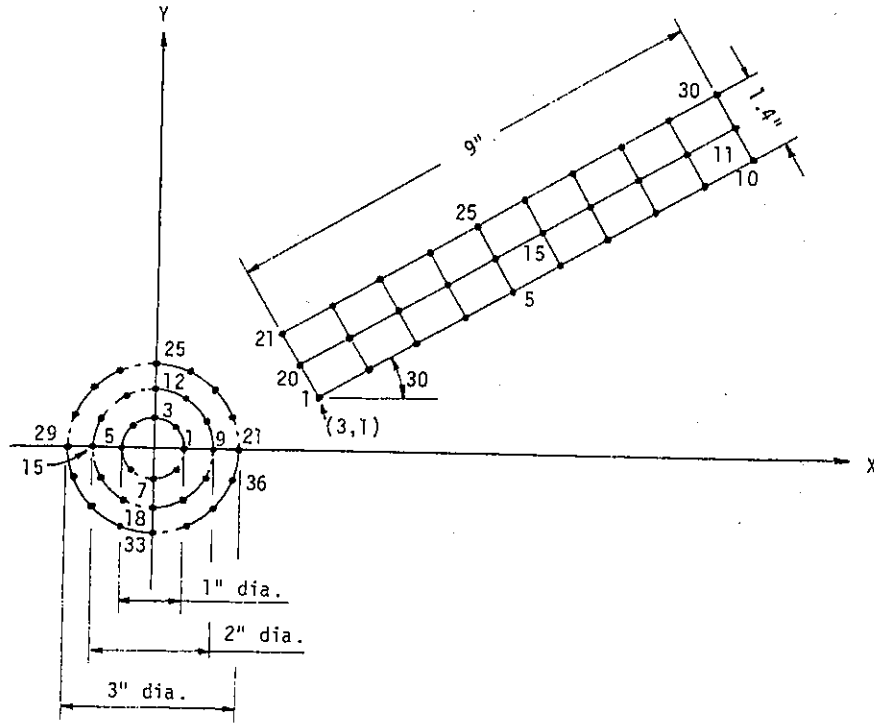
PROBLEMLER

- 5.1 Mümkün olan en kısa ve öz PATTERN deyimini kullanarak, Şekil P5-1'deki iki nokta grubunun her birini dizi olarak tanımlayınız. Tüm noktalar $z = 0$ düzleminindedir.
- 5.2 Problem 5-1'de yer alan ARC ve LINEAR dizilerindeki noktaların, sırasıyla $z = 1.0$ inç ve $z = 2.0$ inç düzeylerindeki delikleri temsil ettiğini varsayınız. ARC (Çember yayı) dizisindeki deliklerin derinliği 0.5 inç; LINEAR (lineer) dizideki delik derinliği ise 0.6 inçtir. Delme ilerleme hızı 2.0 inç/dak.'dır. Başlangıç ve bitiş noktaları (0,0,3)'tür.
- a. Bu iki nokta grubunu bir dizi olarak tanımlayan
b. Delme işlemini tanımlayan
bir APT program parçasını yazınız.
- 5.3 Bir nokta dizisinin:
biçiminde tanımlandığını ve, PO'nun (0,0,0) noktasını simgelediğini varsayınız. Aşağıdaki dizilerde

PTO=PATTERN/(1,1), (0,1),(-1,0),(0,-1)

tanımlanan noktaların konumlarını ve sıralarını belirleyiniz ve CLDATA çıktısı vasıtasıyla bunları doğrulayınız.

- a. PT1=PATTERN/PTO,ATTACH,PO,TRANPT,(3,3)
b. PT2=PATTERN/PTO,ATTACH,PO,TRANPT,(3,3,1)
c. ZSURF/1,0; PT3=PATTERN/PTO,ATTACH,PO,TRANPT,(3,3)
d. PT4=PATTERN/PT1,ATTACH,PO,INVX,ATTACH,PO,INVY
e. PT5=PATTERN/PT1,ATTACH,PO,INVX,INVY
f. PT6=PATTERN/PTO,PT1
g. PT7=PATTERN/PTO,INVERS,PT1
h. PT8=PATTERN/PTO,PT1,INVERS
- Not= CLDATA çıktısını elde etmek için, hareket deyiminden önce CLDATA deyimini ilave ediniz.
- 5.4 Çeşitli dizi deyim formatları, karmaşık bir nokta dizisini tanımlamak üzere sıraya konulabilir. Aşağıda Şekil P5-4'de gösterilen P1,P2,P3,....,P8 noktalarını kapsar. PAT1 nokta dizisini tanımlamak için kullanılan bazı deyimler görülmektedir:
- a. PT1=PATTERN/(3,1,5),(4,1,5)
PT2=PATTERN/(4,2,5),(4,3,5)
PT3=PATTERN/AT,(4,3,5),RTHETA,CONST,3,0,27,54,81
PAT1=PATTERN/PT1,PT2,PT3

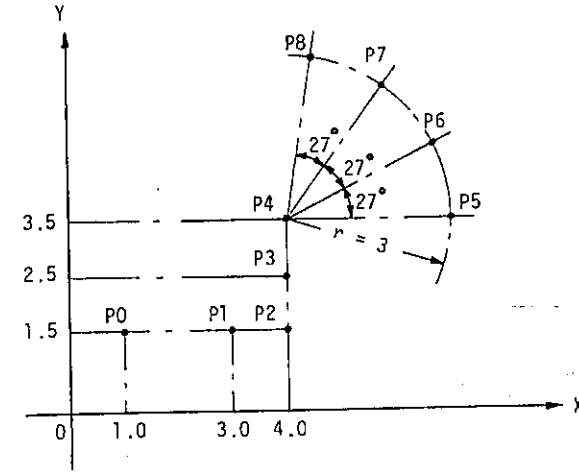


Şekil P5-1

- b. PAT1=PATTERN/AT,(1,1.5),DELTA,XCOORD,2,1,YCOORD,1,1,AT,\$
(4,3,5),RTHETA,CONST,3,0,27,54,81
- c. PAT1=PATTERN/AT,(1,1.5),DELTA,XCOORD,2,1,YCOORD,1,1,\$
RTHETA,CONST,3,0,27,54,81
- d. PAT1=PATTERN/AT,(1,1.5),ABSLTE,XCOORD,3,4,YCOORD,2,5,3,5,ARC,\$
(CIRCLE/4,3.5,3),ATANGL,0,CCLW,81,NUMPTS,4
- e. PAT1=PATTERN/(3,1.5),(4,1.5),DELTA,YCOORD,1,1,ARC,\$
(CIRCLE/4,3.5,3),ATANGL,0,CCLW,81,NUMPTS,4

Bu deyimleri doğrulayınız.

- 5.5 APT deyimlerini kullanarak, Şekil P5-5'de (a'dan g'ye) verilen başlangıç hareketlerini tanımlayınız. Başlangıç noktasının P1 olduğunu ve verilen sembollerle geometrik cisimlerin tanımlanmış olduğunu varsayınız.
- 5.6 APT deyimlerini kullanarak, Şekil P5-6'da (a'dan f'ye) verilen çevresel işleme hareketini tanımlayınız. Kesici takımın, çevresel işleme hareketini başlatmak için gerekli P1 konumuna tam olarak yerleştiğini varsayınız. Verilen sembollerle geometrik cisimlerin de tanımlandığını varsayınız. Pozitif X yönünün şekil sol

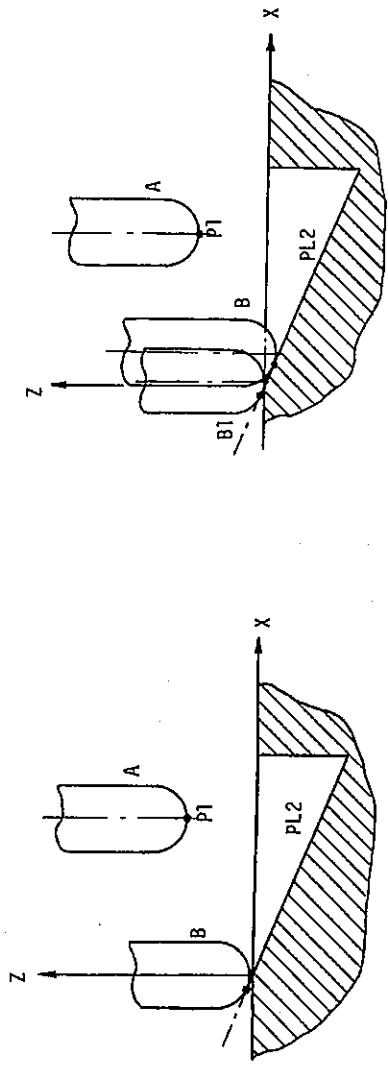


Şekil P5-4

tarafından sağına doğru olduğunu da varsayınız.

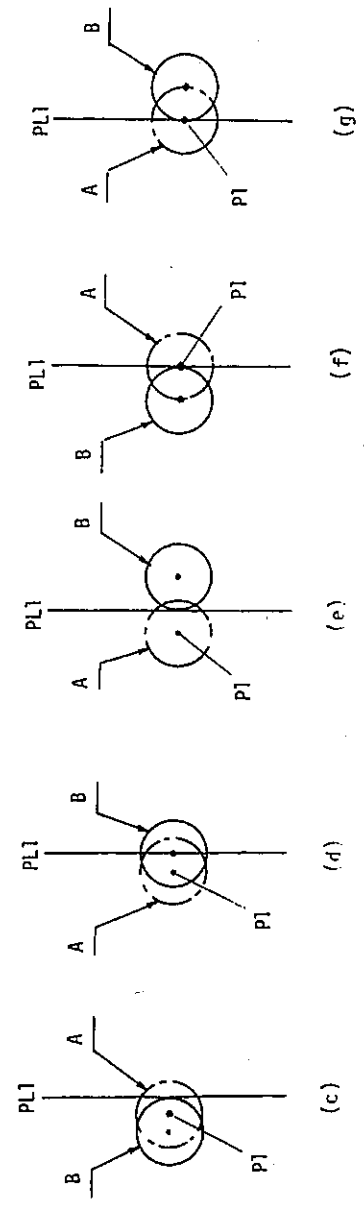
- 5.7 Problem 4.2.'de verilen şablon profilini tezgahta işlemek için gerekli kesici yolunu tanımlayınız. Kesicinin başlangıç ve bitiş konumlarının (6,3,1), ve şablon kalınlığının 0.15 inç olduğunu varsayınız. Koordinat sistemi, Şekil P4-2'de gösterildiği gibi tanımlanmıştır. Kesici hızı 2 inç/dak.'dır.
- 5.8 Profili doğru, çember, çizelgelenmiş silindirlere ve/veya LCONICS'lerden oluşan ve problem 4.4'de verilen parçayı işlemek için gerekli kesici hareket yolunu tanımlayınız. Kesicinin başlangıç konumu (-5,5,1), parçanın kalınlığı ise 0.5 inç'tir. Koordinat sistemi, Şekil P4-4'deki gibi başlangıç noktası parçanın yüzeyi üzerinde olacak şekilde tanımlanmaktadır. 2 inç/dak.'lık ilerleme hızı kullanılmaktadır.

Not: Başka bir geometrik cisme teğet olması esasına göre yapılan bir LCONIC veya çizelgelenmiş silindir tanımı; çevresel işleme hareketi deyiminde bir TANTO değiştiricisinin; kesici takımını bir LCONIC'den ya da çizelgelenmiş silindirden, kendisine teğet olan bir geometrik cisme doğru ya da bu cisimden LCONIC veya çizelgelenmiş silindire doğru hareket ettirmek için başarıyla kullanılacağını garanti etmez. Bunun nedeni; iki geometrik cismin teğet noktasındaki eğrilik yarıçaplarının, o noktada eğimleri aynı olsa bile, farklılık gösterebilmesidir. Bir çözüm; teğet noktasından geçen ve ortak teğet doğrusuna dik olan bir doğru tanımlamaktır. Bu durumda doğru, ON değiştiricisiyle birlikte, kesici takımını iki geometrik cismin teğet noktasına konumlamak için kontrol yüzeyi olarak

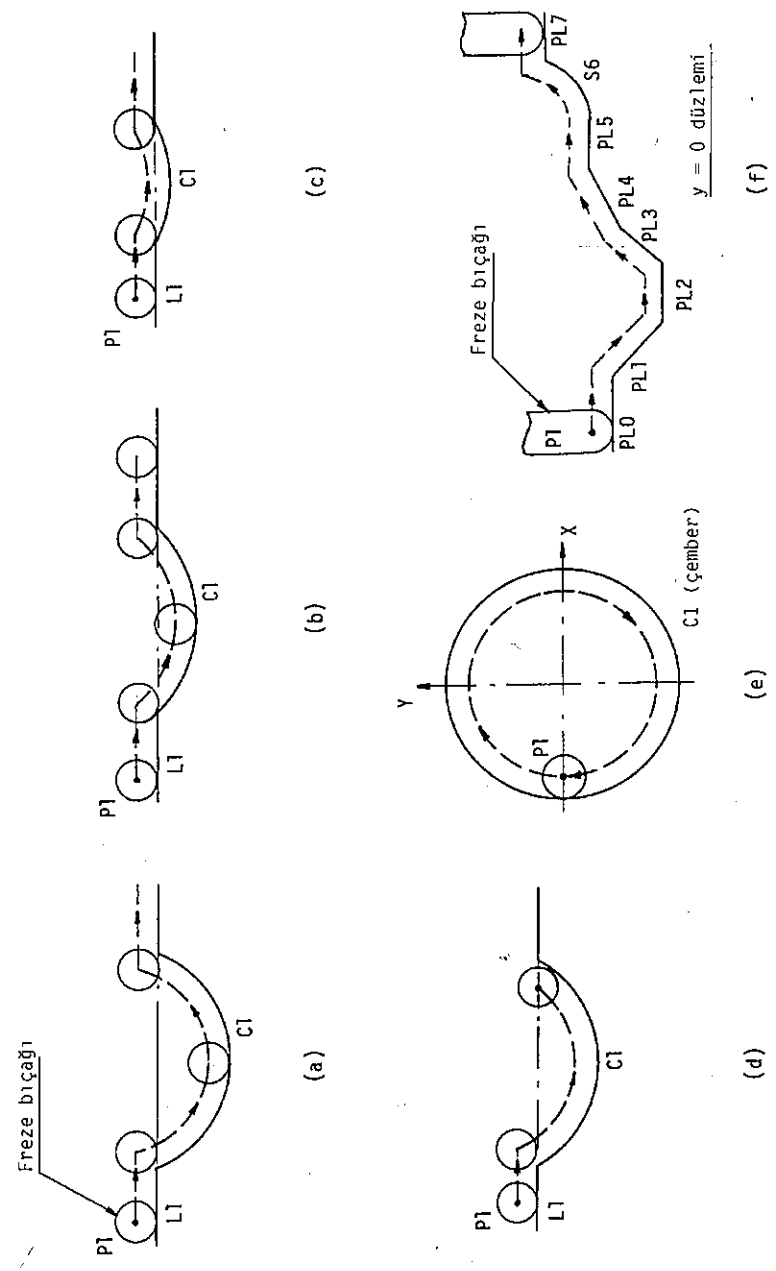


y = 0 düzlemi (a)

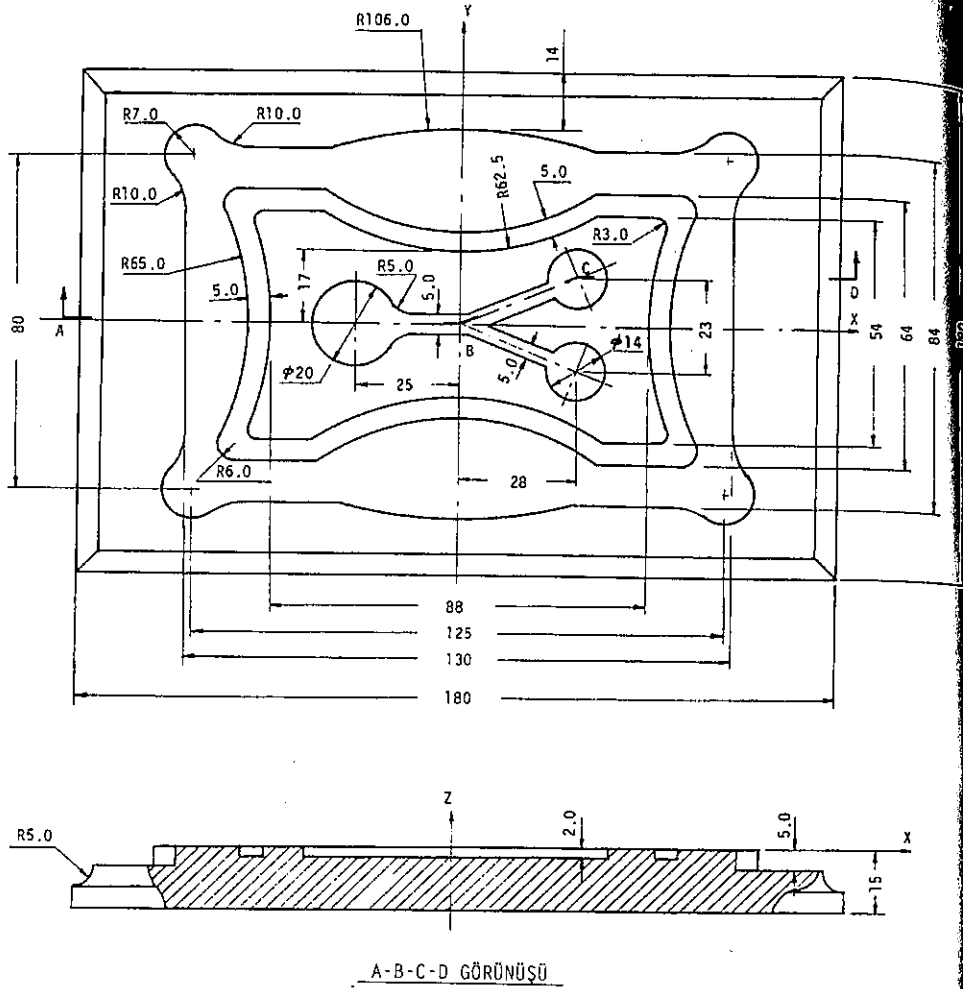
y = 0 düzlemi (b)



Şekil P5-5 Başlangıç hareketleri. (c) den (g)'ye kadar olan şekillerdeki pozitif X yönü (a)'da gösterilenle aynıdır. A: Kesicinin başlangıç konumu; B ve B1: Başlangıç hareketinin bitiş konumları.



Şekil P5-6



Şekil P5-11

kullanılır. Bazı durumlarda, teğet noktasında bir eğriden diğerine eğrilik yarıçapındaki ani değişim nedeniyle, çevresel işleme hareketi, kesici takımın bir sonraki çevresel işleme hareketi için gerekli konuma hareket ettirmek için iki çevresel işleme hareketi deyimleri arasında bir başlangıç hareketi deyimini yerleştirilmedikçe devam etmeyebilir.

- 5.9 Eğer Problem 5.8'de incelenen parçanın frezede işlenmesi için, biri kaba ve diğeri de ince (son) işleme olmak üzere iki paso gerekli ise; kaba işlemeden hemen sonra geri dönüş hareketi olmaksızın ince (son) işlemeye geçilmesini sağlayacak bir APT program parçası yazınız. Hassas işleme payı 0.02 inç'tir.

- 5.10 Şekil P4-8'de gösterilen matrisin boşluğu üç-eksenli bir freze tezgahında açılacaktır. Kesicinin hareket yolu; B-C ve C-D kesit düzlemlerine paralel olan düşey düzlemlerin, kesici hareketini soldan sağa veya sağdan sola doğru, bitiş yüzeyleri boyunca bir düşey düzlem grubundan, bir sonrakine artışı olarak (örneğin D'den E'ye hareket) yönlendirmek üzere kontrol yüzeyi olarak kullanılabilmesini sağlayacak şekilde tasarlanmıştır. Buna göre, kesici hareketi $B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \dots$ 'dir.

Başlangıç ve $B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G$ çevresel işleme hareketini tanımlayan bir APT programı parçası yazınız. Kesicinin başlangıç ve bitiş konumları (10,5,3), ilerleme hızı ise 2.0 inç/dak.'dır. Küresel uçlu bir freze kesicisi kullanıldığını varsayınız.

- 5.11 Şekil P5-11'de verilen parçanın frezede işlenmesi için bir APT programı gerekmektedir. Ham blok malzemenin boyutları 180 X 120 X 15 mm olup; şu üç kesici kullanılmaktadır:

- Kavisli kenarını işlemek için 10 mm'lik küresel uçlu ($R = 5$ mm) freze bıçağı
- 5 mm derinlikteki dış profili frezelemek için 8 mm'lik alın freze
- Kanal ve dairesel boşlukları (2 mm derinlikte) açmak için 4 mm'lik düz uçlu parmak freze

Geometrik cisimleri tanımlayınız ve her biri bir kesici hareketini tanımlamak üzere, üç program parçası yazınız. Başlama ve bitiş noktaları (200,150,30); ilerleme hızı ise 50 mm/dak.'dır.

Bölüm 6

İşleme Özellikleri ve Çeşitli Deyimler

Bir APT Programcısının bakış açısından, bir işleme yöntemi genel olarak, kesicinin takip etmesi gereken yol ve iş parçasını üretmek için gerekli işleme özellikleri olmak üzere iki ana durumu içine alır. İşleme özellikleri; örnek olarak tolerans, talaş payı, kesme hızı, ilerleme hızları, kesici alet ve soğutma sıvısını içerir. Bir APT programı bir NC programını elde etmek için bir bilgisayar tarafından işlenebilen bilgisayar dilinde yazılmış işleme yönteminin tanımlanmış şeklidir. Böylece, bu APT Programı işleminin nasıl yapılacağını belirten talimatları da içermelidir. Bu bölümde, işleme özellikleri ve bilgisayar işlemlerini tanımlayan deyimler tartışılacaktır. Bu deyimlerin bazıları son işlemci tasarımına da bağlıdır. Ayrıca, bunların deyim yapısı ve parametreleri seçilen son işlemciye bağlı olarak değişir. Bundan dolayı doğru bir program yazmak için, programlayıcının kullanılan son işlemci hakkında bilgi sahibi olması gerekir.

6.1 TOLERANSLARI TANIMLAYAN DEYİMLER

Bir NC tezgahındaki kesicinin hareketi, X,Y ve Z yönlerindeki hareketleri üreten motorlara NC denetleyicisi tarafından gönderilen sinyaller vasıtasıyla kontrol edilir. Herhangi bir anda, herhangi bir yöndeki kesici hareketi motorlara gönderilen sinyal oranı tarafından tespit edilir ve istenilen kesici yolu bir eğriyi oluştursa bile bu hareket doğrusaldır. Benzer şekilde, NC işlemcisi, bir çevresel yüzey için kesici aletin yolunu doğrusal enterpolasyona bağlı olarak hesaplar (Şekil 5-27). Bundan dolayı, istenilen kesici yolundan bir miktar sapma olması kaçınılmazdır. Programcı iş parçasının tasarımına göre bu sapmaya kabul edilebilir bir sınır tanımlamalıdır. Bu sınır toleans olarak adlandırılır. APT'de tolerans aşağıdaki deyimlerle tanımlanabilir:

INTOL/t

birimi milimetre veya inç olabilen t , tasarlanmış profilden kesici konumunun izin verilebilir en büyük içe doğru sapmasıdır. (Şekil 6-1 (a)).

OUTTOL/t

burada t , tasarlanmış profilden kesici alet konumunun izin verilebilir en büyük dışa doğru sapmasıdır (Şekil 6-1 (b)); ve

TOLER/t

buradaki t dışa doğru toleransı temsil eder ve içe doğru tolerans sıfırdır (Şekil 6-1 (b)).

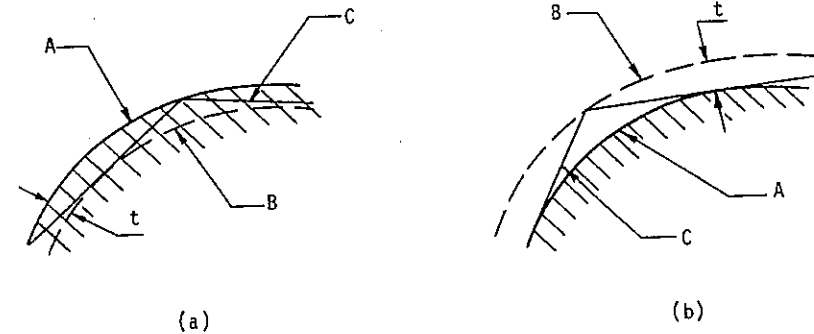
İş parçası tasarımında ortaya çıkabilecek muhtemel üç durum vardır. Bunlar tasarlanmış gerçek profilden izin verilen sapma; içeri doğru, dışarı doğru veya her iki yönde birden olması durumudur. İçeri doğru tolerans INTOL deyimleriyle dışarı doğru tolerans ise OUTTOL veya TOLER deyimleriyle tanımlanır. Her iki yönde de toleransları belirtmek için INTOL ve OUTTOL deyimlerinin ikisi de tanımlanmalıdır.

Yukarıda anlatılan deyimlerde tanımlanan tolerans t , tahrik, parça ve kontrol yüzeylerine eşit şekilde tatbik edilir. Eğer, üç farklı denetim yüzeyi için farklı toleranslar gerekirse aşağıdaki deyimler kullanılabilir:

$\left[\begin{array}{l} \text{INTOL} \\ \text{OUTTOL} \\ \text{TOLER} \end{array} \right] / t1 [,t2[,t3]]$

burada $t1$, $t2$ ve $t3$ sırası ile iş parçası, tahrik ve denetim yüzeylerindeki toleranslardır. Tahrik yüzeyi üzerinde tolerans $t3$ kullanılmazsa, tahrik yüzeyi üzerinde tolerans denetim yüzeyi üzerindeki tolerans $t2$ alınır.

Yukarıda verilen tolerans komutları modaldır; yani, diğer bir tolerans deyimini tarafından değiştirilinceye kadar geçerlidirler.



Şekil 6-1 (a) Kesici yolu (C) ve INTOL/t deyimini tarafından tanımlanmış kabul edilebilir profil (B). (b) Kesici yolu (C) ve OUTTOL/t veya TOLER/t deyimini tarafından tanımlanmış kabul edilebilir profil (B). A: Tasarlanmış iş parçası profili.

6.2 ÖLÇÜ BİRİMİNİ TANIMLAYAN DEYİM

Teknik resimlerde bir ölçüyü tanımlamak için iki temel ölçü birimi kullanılır. Bunlar: milimetre (mm) ve inç (in). dir.

UNITS / $\left\{ \begin{array}{l} \text{[INCHES]} \\ \text{[MM]} \end{array} \right\}$

Bu deyim kullanılan birimi göstermek için kullanılabilir. İnç varsayılan birimdir ve tanımlanmayabilir. Eğer, bir programda birden fazla UNITS deyimini tanımlanmışsa, herbiri tanımlanan bir önceki birimi iptal eder ve bu yeni tanımlanan birim kendinden sonra gelen program bölümünü etkiler.

Ölçme birimlerinin dönüşümü sadece geometrik öğelere ve kesici yerleşimine atbik edildiği skalar değişkenler bundan etkilenmez. Örneğin, aşağıdaki programda

```

.....
UNITS/MM
I = 1
P1 = POINT/3, 4,5
UNITS/INCHES
P2 = POINT/I (I+1), (I+2)
.....
A) GOTO/P1
B) GOTO/P2
.....

```

P1 noktası mm cinsinden tanımlanmıştır ve bunun x, y, z koordinatları 3,4 ve 5 mm dir.

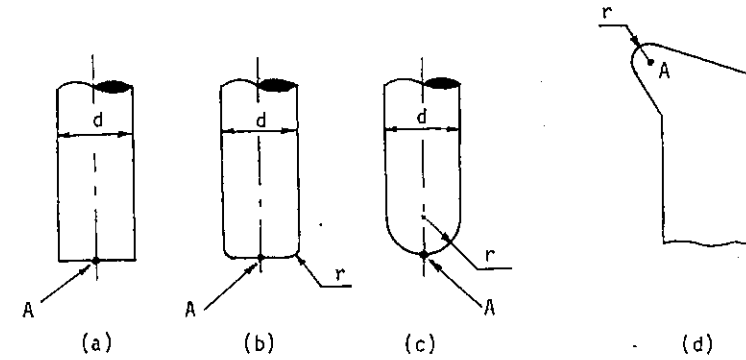
I harfi bir skalar değeri temsil eder ve UNITS/MM deyimini tarafından etkilenmez. Böylece, P2 noktasının x, y ve z koordinatları sırası ile 1,2 ve 3 inç'tir. Bu program bölümündeki kesici yolu UNIT/INCHES deyiminden sonra tanımlanmıştır. Deyimler tarafından tanımlanmış olan kesici konumu: sırası ile (3/25.5, 4/25.4, 5/25.4 inç) ve (1,2,3 inç) tir.

Fakat gerçekte APT sistemi ölçüsüzdür. Geometrik öğeleri veya kesici alet yolunu tanımlamak için bir programda farklı ölçü sistemlerini kullanma niyeti yoksa, ölçü birimini tanımlamaya gerek yoktur. Bu durum sadece geometrik öğelerin farklı birim sistemlerinde tanımlanması ihtiyacı duyulduğunda hesaplama zamanını azaltmak için faydalıdır.

6.3. KESİCİ ALETİ TANIMLAYAN DEYİMLER

6.3.1. CUTTER Deyimi

Daha önce de işaret edildiği gibi NC işlemcisi tarafından işlenen kesici yerleşim verisi (CLDATA), kesici eksen çizgisi konumlarını veya daha belirli bir ifade ile bir freze kesicisi (çıkışı) için uç merkez konumu veya bir torna kesicisi (kalemi) için kesici ucu merkezinin konumunu temsil eden bir dizi koordinat listesidir. (Şekil 6-2).



Şekil 6-2 A noktası, bir kesicinin konumunu temsil eder. (a) Düz uçlu parmak freze; (b) köşesi kavisli parmak freze; (c) yuvarlak uçlu freze çakışı; (d) Torna kalemi.

Noktadan - noktaya hareket için bir kesici aletin geometrik şekli önemli değildir. Çünkü kesicinin eksen çizgisi konumu noktadan - noktaya hareket deyiminde önceden tanımlanmıştır. Fakat, 5. Bölümde de tartışıldığı gibi, bir çevresel hareket, çevresel hareket esnasında yer alan ve birini takip eden konumlarda kesici alet profili iş parçası profiline bağlı olarak tanımlanır. NC işlemcisi, iş parçası profiline, tanımlanan toleransa ve kesicinin geometrik biçimine göre kesici eksen çizgisinin konumunu hesaplar. Kesici eksenine dik olan düzlemde (yani x - y düzlemi) bir parmak freze ile yapılacak ferezeleme işlemi için sadece kesici çapı, d, ve köşe kavis yarı çapı, r, kesici merkezinin konumunu hesaplamak için gereklidir. (Şaekil 6-2). Aşağıdaki deyim parmak frezeleri tanımlar:

CUTTER/d [,r]

burada d ve r sırasıyla kesici çapı ve köşe kavis yarı çapıdır (Şekil 6-2). Bunlar pozitif olmalıdır. Eğer sıfır ise r parametresi kullanılmayabilir (düz uçlu parmak frezede olduğu gibi).

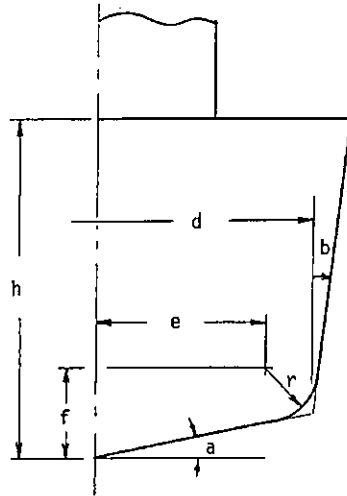
Bir torna kalem için kesme işlemi kesicinin uç noktası tarafından gerçekleştirilir. Böylece torna kalem de çapı $2r$ olan bir düz uçlu parmak freze olarak düşünülebilir (Şekil 6-2)

Üç boyutlu işleme için, bir parça profili APT de izin verilen yüzeylerin herhangi bir birleşimi olabilir. Kesici eksen çizgisi konumunun hesaplanması, kesicinin üç boyutlu çevresinin tanımlanmasını da gerektirir. Bu amaçla bir kesiciyi tanımlamak için ek parametrelere ihtiyaç vardır. Üç boyutlu işleme için bir kesiciyi tanımlayan deyim, aşağıda verildiği gibidir (Şekil 6-3):

CUTTER/d,r,e,f,a,b,h

Burada::

- d = kesicinin uç ve kenar yüzeylerinin kesişmesinin oluşturduğu çemberin çapı
- r = köşe çemberinin yarıçapı
- e = köşe çemberi merkezinden kesici eksen çizgisine olan uzaklık
- f = köşe çemberi merkezinden kesici eksen çizgisine veya eksenine dik düzleme olan uzaklık.
- a = derece cinsinden, kesicinin uç yüzeyi ve kesici eksenine dik olan düzlem arasındaki açı (açı aralığı $0 \dots a \ 90^\circ$ dir).
- b = derece cinsinden kesicinin yan yüzeyi ve kesici eksen arasındaki açı. (Şekilde gösterildiği gibi pozitif yönde $-90^\circ < 90^\circ$ dir).
- h = kesicinin uç merkezinden ölçülen kesici yüksekliği.



Şekil 6.3 Kesiciyi tanımlayan parametreler (d,r,e,f,a,b,h)

Köşe çemberinin uç ve yan yüzeylere teğet olması gerektiğine dikkat ediniz. Ayrıca, b hariç diğer tüm parametreler, pozitif olmalıdır. *Genellikle r,e,a,b ve h'nin değerleri bilinir. d ve f parametreleri, aşağıdaki formüllere göre bilgisayar tarafından hesaplanmalıdır^{1,2}:

$$d = \{ 2e + r \cdot \cos [45 + (b-a)/2] / \cos [45 - (a+b)/2] \} \dots$$

$$f = r \cdot \sin [45 + (b-a)/2] / \cos [45 - (b+a)/2] + (d/2) \cdot \tan (a)$$

Kesici tanımlama deyimini kullanılmadan önce bu formüllere dayalı iki hesaplama deyimini tanımlanmalıdır.

Yukarıda bahsedilen sınırlamaların yanı sıra, CUTTER deyimindeki parametrelerin değerlerinde bazı sınırlamalar da vardır:

1. $(a+b) < 90^\circ$
2. $h \geq (d/2) \tan (a)$
3. $f \geq (d/2) \tan (a)$
4. Köşe çemberi uç yüzeye teğet olmadığı zaman uç yüzeyden daha aşağıya inmemelidir.

CUTTER/d [,r]

deyimini üç boyutlu işleme için kullanıldığı zaman aşağıda varsayılan değerler NC işlemcisi tarafından kesici aleti tanımlamak için kullanılır:

$$a = 0$$

$$b = 0$$

$$f = \{ 0^r \text{ (Eğer r verilmemişse)} \}$$

$$e = (d/2) - r$$

$$h = \begin{cases} 5 \text{ eğer } r \leq 5 \\ r \text{ eğer } r > 5 \end{cases} \text{ veya } r \text{ verilmemişse}$$

Şekil 6-4 CUTTER deyiminin birkaç farklı kullanımını vermektedir.

6.3.2 Kesici Üst Yüzeyinin Kesici Yoluna Etkisi

Bir çevresel işleme hareketinin kesici yolu, kesici ucu ve kesici yan yüzeylerinin üç denetim yüzeyleri ile (parça, tahrik ve kontrol yüzeyleri) etkileşimi tarafından tespit edilir. Üç denetim yüzeyinden herhangi biri kesicinin üst yüzeyini sınırlayan bir yüzey olmadığı sürece bu durum geçerlidir. Genellikle, bu üç denetim

* Bir iki şart APT kullanımı kılavuzunda açıklanmıştır. 1,2. Fakat, e parametresi de negatif olabilir. (yani köşe çemberi ve onun merkezi eksen çizgisinin farklı tarafındadır). Ayrıca köşe çemberi uç ve yan yüzeylere teğet olmayabilir.

yüzeyi, kesicinin üst yüzeyinde içine alan kesici profili için sınırlama yüzeyleridir. Bundan dolayı parça profilinin bir kısmı olan bir geometrik öge kesici eksenine ile birden fazla kesişme noktasına sahipse kesici yolunun tanımlanmasında kesici aletin yüksekliği de hesaba katılmalıdır. Örneğin, Şekil 6-5'te gösterilen A-B-C-D kısmı, herbirinin silindir olarak tanımlandığı bir kaç dairesel yaydan oluşmaktadır.

Aşağıdaki deyimler istenilen kesici yolunu, PO-A-C, tanımlamak için kullanılırlar:

CUTTER/1,0,0,0,0,0,3 \$\$\$ D = 1; H=3

FROM/PO

INDIRV/0,0,-1

GO/ON,PL1,CYL1,ON,PL2 \$\$\$PL2 IS X-Z PLANE AND PL1 IS PLANE X=T

INDIRV/-1,0,0

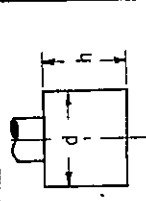
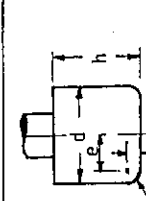
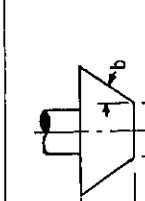
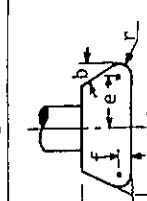
TLON,GOFWD/PL2,PSTAN,CYL2

Ancak, CYL1 yüzeyi CYL2 yüzeyine teğet olduğundan kesici C noktasına ulaşamaz çünkü kesicinin B noktasına ulaşmasından sonra silindirin (CY) üst kısmı etkilenir. Kesici yolu PO-A-C yerine PO-A-B-C' olacaktır. Z yönünde birden fazla levhalı elitik koniler ve hiperbolik silindirler gibi yüzeyler de bazen kesici konununun tanımlanmasında probleme neden olurlar (Şekil 6-6). Bunun için, kesici yolu tanımlanırken iş parçası profilini oluşturan geometrik öğelerin dış sınırları yüzeylerin üst kısmı ile kesicinin birbirine muhtemel girişimleri tesbit edilebilecek şekilde hesaba katılmalıdır.

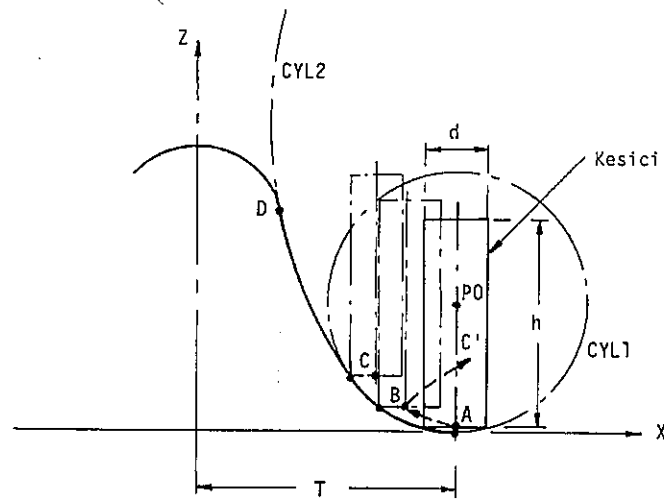
Yukarıda açıklanan problem için bir kaç çözüm yolu vardır.. Birincisi, kesici eksenine ile sadece bir kesişmesi olan bir geometrik öge olarak yüzeyi yeniden tanımlamaktır. Örneğin, CYL1 ve CYL2 silindirlerinin AC ve CD kesitleri sıralanmış bir silindir olarak tanımlanırsa, kesici alet A'dan C'ye hareket eder. Çünkü yeniden tanımlanan bu geometrik öğeler, kesici eksenine ile birden fazla kesişmeye sahip değildirler.

Diğer bir yaklaşım da, kesici yolu hesaplanırken kesici yüksekliğini teorik olarak azaltmaktır. Örneğin, Şekil 6-6 (a) daki kesici yüksekliği h'den az ise kesici üst yüzeyi hevhası S1 ile girişimi olmaksızın A noktasına ulaşabilir. Elbette, CUTTER deyimini kesici yüksekliğini geçici olarak değiştirmek için kullanılabilir, fakat kesici yüksekliği CUTTER deyimini için kabul edilemeyecek bir değere indirilmesi gerektiğinde bu durum yine problem çıkarabilir. Bu problemi çözmek için APT'de ek bir deyim vardır. Bu deyim formatı aşağıdaki şekildedir.

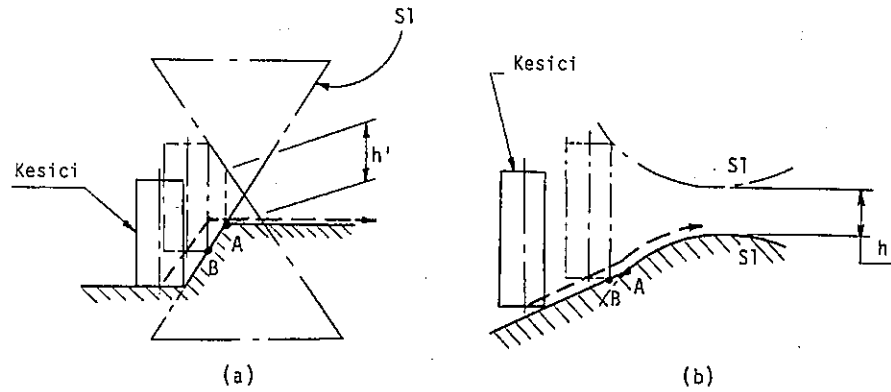
CUTTER / OPTION', $\begin{Bmatrix} 1 \\ 2 \end{Bmatrix}$ $\begin{Bmatrix} r, h \\ OFF \end{Bmatrix}$

Kesici	Deyim	
	Basit biçim	Genişletilmiş biçim
	CUTTER/d veya CUTTER/d,r	CUTTER/d,0,d/2,0,0,0,h
	CUTTER/d,r	CUTTER/d,r,e,f,0,0,h
		CUTTER/d,0,d/2,0,0,b,h (b > 0)
		CUTTER/d,r,e,f,0,b,h (b < 0)

Şekil 6.4 CUTTER deyiminin örnekleri



Şekil 6.5 Denetim yüzeylerinin Z (kesici) eksenine göre kesilmesi durumunda kesici alet yüksekliği hesaba katılmalıdır.



Şekil 6.6 Z yönünde iki levhalı yüzeyler, kesiciyi konumlandırma problemlerine sebep olabilir. S1 eliptik koni (a) veya hiperbolik silindir (b). B: Kesici alet bu noktada yüzey ile girişime uğrar. -----> ; istenilen kesiti yolu.

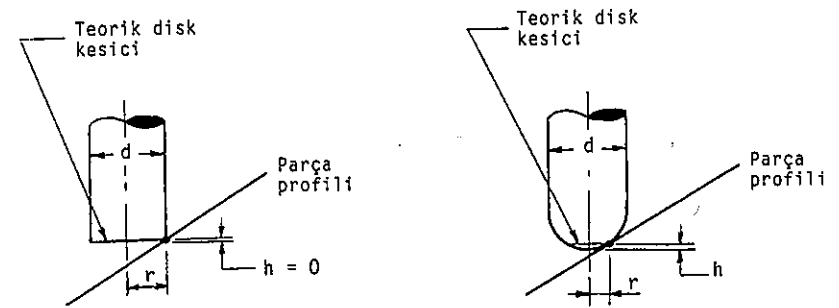
Kesici çevresinin iş parçası profiline teğet parametre r ve h kullanır (Şekil 6-7). Bu konumda gerçek kesiciyi değiştirmek için sıfır yükseklik ve r yarıçaplı, bir teorik disk kesici varsayılır. Bu teorik disk kesici ile gerçek kesici uç merkezi arasındaki mesafe h olduğu için, gerçek kesici konumu teorik kesicinin konumuna bağlı olarak NC işlemci tarafından hesaplanabilir. 1 ve 2 parametreleri teorik kesicinin parça ve tahrik yüzeylerini kullandığını gösterir. Gerçek kesici tekrar yürürlüğe girdiği zaman teorik kesici, diğer bir CUTTER/OPTION deyimini veya aşağıdaki deyim tarafından iptal edilebilir.

CUTTER/OPTION, {1}, OFF

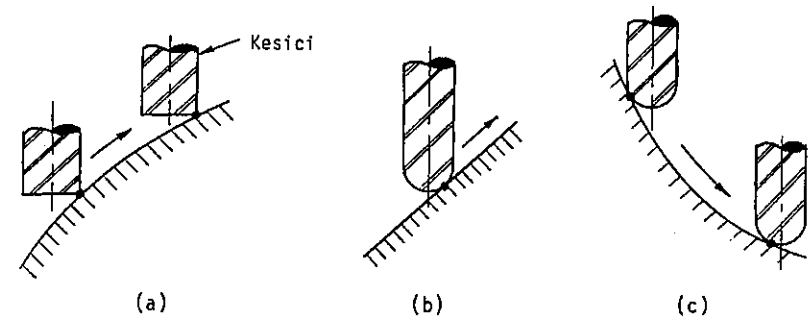
Yukarıda belirtilen ifadeden anlaşılacağı üzere, esas itibari ile CUTTER/OPTION Deyimi kesici ve parça profili arasındaki temas noktasını tanımlar.

Eğer kesicinin düz bir ucu varsa (Şekil 6-8 [a]) veya eğer parça profili düzlem yüzey veya bir doğru ise (Şekil. 6-8 [b]) kesicideki temas noktasının konumu değişmez. Bunun için düz uçlu bir kesici kullanıldığında veya doğrusal bir profil işlendiğinde CUTTER/OPTION deyimini bir probleme neden olmaz.

Fakat, doğrusal olmayan bir profili işlemek için düz uçlu olmayan kesiciler kullanıldığı zaman teorik kesiciye göre hesaplanan bir çevresel hareketin kesin



Şekil 6.7 CUTTER/OPTION deyimini tanımlayan temas noktası veya teorik disk kesici.



Şekil 6.8. Kesici ile parça profili arasındaki temas noktası (a) talaş kaldırma anında düz uçlu kesicinin temas noktasının kesici üzerindeki konumu değişmez. (b) parça profili bir doğru parçası veya düzlem yüzey ise talaş kaldırma anında profil uçlu kesicinin temas noktasının kesici üzerindeki yeri değişmez, (c) parça profili doğrusal değilse temas noktası kesici üzerindeki konumunu değiştirir.

konumları doğru olmayabilir. (Şekil 6-8 [c]). Kesicinin üst yüzeyi ile parça veya tahrik yüzeyinin girişiminden kaçınmak için bir diğer muhtemel yaklaşım ise, xy düzleminde kesici yolunu tanımlamak ve bunu istenilen kesici yoluna sonradan dönüştürmektir. Bu yöntem 7. Bölümde ele alınmaktadır.

CUTTER/OPTION deyiminin kullanımını açıklamak için aşağıda iki örnek verilmiştir

Örnek 1

Aşağıdaki program parçasında olduğu gibi kesici yüksekliği azaltılırsa Şekil 6-5 te gösterilen parçayı işlemek için önceden verilen program kullanılabilir:

```

.....
CUTTER/1,0,0,0,0,0,0,3
FROM/PO
INDIRV/0,0,-1.
GO/ON,PL1,CYL1,ON,PL2 $$ CYL1 IS THE PART SURFACE
CUTTER/OPTION,1,0.5,0
INDIRV/-1,0,0
TLON,GOFWD/PL2,PSTAN,CYL2
CUTTER/OPTION,1,OFF
.....

```

Programın bu kısmı için kesici yolu PO - A - C dir. CUTTER/OPTION deyiminde parametrenin 1 seçilmesinin sebebi, kesici ile girişimi olan yüzeyin parça yüzeyi olmasıdır (CYL 1).

Örnek 2

Teorik kesicinin tahrik yüzeyine uygulanması Şekil 6-9'da görülebilir. Bu durumda A noktasından B aracılığı ile C noktasına kesiciyi hareket ettirmek için aşağıdaki deyimler kullanılabilir: \$\$ TOOL IS AT POINT

```

..... $$TOOL IS AT POINT A; PART SURFACE IS PL2
CUTTER/OPTION,2,0.5,0
INDIRV/-0.2,-1,0
TLRGT,GOFWD/CYL1,PAST,PL1
CUTTER/OPTION,2,OFF
PSIS/PL1
INDIRV/0,-1,0
TLLFT,GOFWD/PL2.....
.....

```

Burada kesicinin üst yüzeyi ile silindir CYL1'in girişiminden kaçınmak için bir teorik kesici kullanılmaktadır.

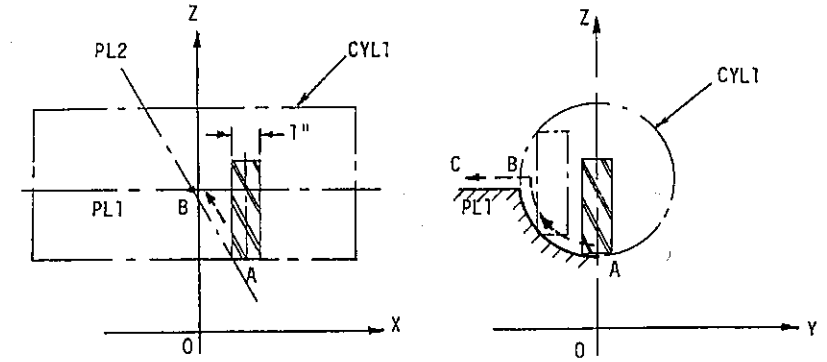
6.4 İLERLEME HIZINI TANIMLAYAN DEYİMLER

İlerleme hızı, parça profili boyunca hareket eden veya bir noktadan diğer bir noktaya hareket eden kesicinin hızına denir. Bir dakikada veya takım tezgahı milinin bir devrinde kesicinin hareket ederek (inç veya milimetre cinsinden) gittiği mesafe olarak tanımlanır. Normal olarak, ilerleme hızı freze tezgahları için dakikada inç (parmak) veya milimetre cinsinden ve torna tezgahları için bir devirde inç veya milimetre cinsinden verilir.

İlerleme hızı aşağıdaki deyim kullanılarak tanımlanabilir:

$$\text{FEDRAT} / \left\{ \begin{array}{l} \text{IPM, I} \\ \text{IPR, J} \end{array} \right\} f$$

burada:



Şekil 6.9. Kesicinin üst yüzeyi tahrik yüzeyi (CYL1) ile girişime uğramakta.

UNITS deyiminden tanımlanan birime bağlı olarak IPM, f ilerleme hızının, dakikada inç veya milimetre cinsinde ölçüldüğünü gösterir. IPR tanımlanan f ilerleme hızının, bir devirde inç veya milimetre cinsinden ölçüldüğünü gösterir. Kesicinin hızlı hareketi; şu deyimle tanımlanır

RAPID

Bir ilerleme hızı deyimini, hareket deyiminden önce tanımlanmalıdır. FEDRAT deyimini. Diğer bir FEDRAT deyimini tarafından değiştirilinceye kadar etkindir. Fakat, RAPID deyiminin geçerliliği son işlemcinin tasarımına bağlıdır.

Genellikle, son işlemci ögle tasarlanır ki, RAPID deyimini bir sonraki hareket deyimine kadar etkin olur. Kitabın bundan sonraki kısımlarında bu programlama kuralı kullanılacaktır.

FEDRAT ve RAPID deyimlerinin kullanımını açıklamak için şekil 6-10'da bir örnek verilmiştir.

```

.....
FEDRAT/2.0
.....
..... } (Feedrate is 2.0)
.....
RAPID
GOTO/P1$$ RAPID FEEDRATE
GOTO/P2$$ FEEDRATE IS 2.0.
FEDRAT/3.0
GO/.....
..... } (Feedrate is 3.0)
.....

```

Şekil 6.10 RAPID ve FEDRAT deyimlerinin kullanımı

6.5 İŞ MİLİ DÖNÜŞ HAREKETİNİ TANIMLAYAN DEYİMLER

İş (Fener) mili dönüş hareketi aşağıdaki parametreler tarafından tanımlanır:

1. Fener milinin dönüşünü başlatmak veya durdurmak için kullanılan ON veya OFF,
2. Saatin aynı yönünde veya saatin tersi yönünde olarak iş mili dönüşünün yönünü tanımlayan CLW veya CCLW
3. Dakikadaki devir sayısına bağlı olarak iş milinin dönüş hızını tanımlayan, s., deyiminin yapısı aşağıda verildiği gibidir.

```

SPINDL/s, {CLW }
             {CCLW }
SPINDL/OFF
SPINDL/ON, {CLW }
            {CCLW }

```

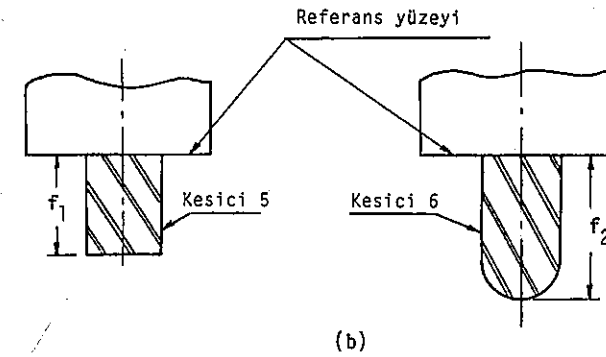
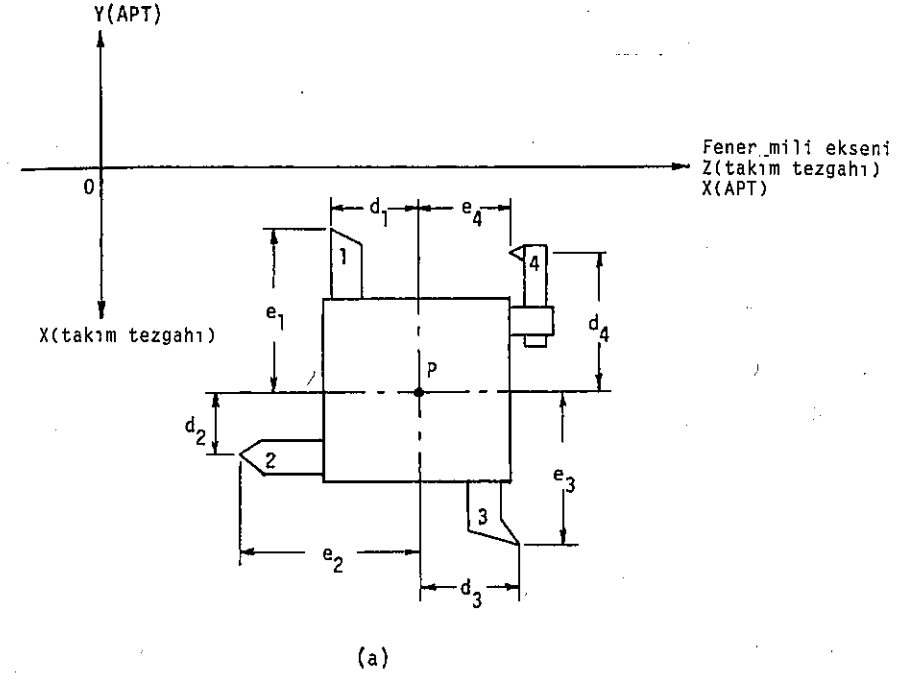
Programın başlangıcında ilk format, s devir sayısı, ve dönüş yönünü tanımlamak için kullanılır. Ayrıca iş milinin dönmesini başlatmak içinde kullanılır. İkinci format ise, iş mili dönüşünü durdurmak için programın ortasında veya sonunda kullanılabilir. Üçüncü format ise, daha önce tanımlanan aynı dönme hızında fener milini döndürmek için programın ortasında kullanılır. İlk format devir sayısını ve dönme yönünü değiştirmek için de kullanılabilir. Eğer dönme yönünde bir değişiklik gerekirse önce iş mili dönme yönünün tersi seçilirse birçok son işlemci iş mili durdurmak üzere tasarlanır. Fakat eğer iş mili dönme yönünün tersi seçilirse birçok son işlemci fener milini durdurma kodunu (örneğin MO5) otomatik olarak üretir. Bu gibi durumda SPIN OL/OFF deyimini kullanılmayabilir.

6.6 KESİCİ TAKIMI DEĞİŞTİRMEK İÇİN KULLANILAN DEYİMLER

Tezgah işlemleri esnasında birçok NC'li takım tezgahı otomatik olarak kesici aletleri değiştirebilir. Tornalarda, tezgahın x ve z yönlerinde farklı kesicilerin farklı

kaydırma değerleri vardır. Freze tezgahlarında kesici eksen daima iş mili eksenidir (mafsallı veya kranklı kesici kullanımı hariç) ve farklı kesiciler için kesici boyutundaki farklılıklar yalnız kesici boyunda ve çapında ortaya çıkar.

Kesici çapı, CUTTER deyimini tarafından tanımlanır. Bundan dolayı istenen kesici ve onun kaydırma miktarını tanımlamak için bir deyim ihtiyacı vardır. Bu deyim yapısı



Şekil 6-11 Takım uçlarındaki farklar. (a) Torna takımı (P: döner kulenin referans noktası). (b) Freze.

$$\text{LOADTL} / n \left[\text{SETOOL} \left\{ \begin{array}{l} d, e, f \\ d, e, f \end{array} \right\} \right]$$

burada n kesici kodudur; d , ve f sırası ile APT koordinat sisteminin X , Y ve Z yönlerindeki kesicinin referans noktasından kesici bağlama aparatının referans noktasına olan kaydırma miktarıdır (Şekil 6-11).

Tornalar için APT koordinat sistemi takım tezgahu koordinat sisteminden farklıdır. Bu 8. Bölümde açıklanacaktır.

Örnek

Şekil 6-11'de gösterilen 1,3,5 ve 6 no.lu kesiciler için LOADTL deyimleri sıra ile aşağıda verilmiştir.

```
LOADTL/1,SETOOL,d1,e1,0
LOADTL/4,SETOOL,d4,e4,0
LOADTL/5,SETOOL,0,0,f1 (veya LOADTL/5,SETOOL,f1)
LOADTL/6,SETOOL,0,0,f2 (veya LOADTL/6,SETOOL,f2)
```

4 no.lu kesicinin kaydırma miktarının onun çalışma konumuna göre tespit edilmesine dikkat etmelidir. (Yani, döner başlık saatin tersi yönünde 90 döndürüldükten sonraki kesici magazinindeki kesici alet seçimi Bölüm 13.4.1.1'de SELCTL deyiminin kullanımı ile programlanır.

6.7 SOĞUTUCUYU TANIMLAYAN DEYİMLER

Bazı NC'li tezgahlarda soğutucunun seçilen yapısı işleme ihtiyaçlarına bağlı olarak farklılıklar gösterir. Soğutucu, fazla sıvı akımı (FLOOD), musluklu soğutucu (TAPKUL) veya pülverize (MIST) olarak kullanılabilir. Soğutucuyu seçmek için, açmak veya kapatmak için kullanılan deyimler aşağıda verilmektedir.

$$\text{COOLNT} / \left\{ \begin{array}{l} \text{FLOOD} \\ \text{MIST} \\ \text{TAPKUL} \end{array} \right\}$$

$$\text{COOLNT} / \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$$

İlk deyim, soğutucuyu seçmek, değiştirmek ve açmak için kullanılır. COOLANT/OFF deyimini programın ortasında veya sonunda soğutucuya kapatmak için programın kullanılır. COOLANT/ON deyimini soğutucuya açmak için kullanılabilir.

6.8

İŞLEME ESNASINDA BEKLEMİYİ TANIMLAYAN DEYİM

Bazen, keskin bir köşe elde etmek için veya bir deliğin tabanında daha iyi bir yüzey kalitesi elde etmek için işleme esnasında beklemeye ihtiyaç duyulur. Belli bir süre için bekleme aşağıdaki deyim kullanılarak tanımlanabilir:

DELAY/t

burada t , saniye cinsinden bekleme süresidir.

Örnek

Delik delme işlemi esnasında, aşağıdaki programda yer alan DELAY deyimini deliğin tabanında bir bekleme oluşturur:

```
.....
.....
FEDRAT/2.0
GODLTA/-1.0    $$ DRILL MOVES DOWN AT SPEED 2.0 IN/MIN
DELAY/0.5      $$ DRILL STOPS AT THE BOTTOM FOR 0.5 SEC
GODLTA/1.0     $$ DRILL MOVES UP
.....
.....
```

6.9 DAİRESEL HAREKETİ TANIMLAYAN DEYİM

5. Bölümde açıklandığı gibi, çevresel kesici yolu NC işlemcisi tarafından bir dizi doğrusal hareket kademeleri, doğrusal enterpolasyon, ile gerçekleştirilir. Bu hareket kademelerinin uç noktalarına karşılık gelen bir grup koordinat dizisi türetilir. Bir son işlemci tarafından bu veriler işlendiği zaman, ilgili G kodları (veya doğrusal hareket kademeleri) türetilir. Bundan dolayı tek bir dairesel hareket için onlarca veya yüzlerce NC deyimleri olabilir. Gerçekte, bir NC tezgahda dairesel hareket tek bir deyimle tanımlanabilir. Tam bir dairesel hareket için G02 ve G03 kodunu üretmek maksadı ile uç konumun koordinatlarını ve dairesel hareketin merkez noktasını kullanan bir son işlemci tasarlanabilir. Böylece hareketin dairesel olduğunu gösteren bilgi NC işlemcisine aktarılmalıdır. Aşağıdaki deyim dairesel hareketi ifade eder:

$$\text{ARCSLP} / \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$$

Aşağıdaki deyimlerde tanımlanan hareketin, NC işlemcisinin son işleme kısmında dairesel enterpolasyon alt programı tarafından işlenebileceğini ARCSLP/ON deyimini gösterir.

ARCSLP/OFF deyimini dairesel enterpolasyonu iptal eder ve hareket verileri doğrusal enterpolasyon alt programı tarafından işlenir. Eğer ARCSLP/ON deyimini tanımlanırsa, hareket deyimindeki tahrik yüzeyi bir daire veya silindir olduğu zaman bir dairesel enterpolasyon NC kodu G02 veya G03 türetilir. Daire veya silindir olmayan tahrik yüzeyli hareket deyimleri için dairesel enterpolasyon alt programı çağrılmaz ve bunun yerine doğrusal enterpolasyon alt programı kullanılır. Böylece, programda

```

.....
.....
ARCSLP/ON
A1) TLRGT, GORGT/C1,L1    $$ C1 IS A CIRCLE
A2) GORGT/L1,L2          $$ L1 IS NOT A CIRCLE OR CYLINDER
.....
.....

```

A1 diye etiketlenen deyim;

NC işlemcisinin son işlemci kısmı tarafından dairesel enterpolasyon hareket olarak işlenir. Diğer yandan A2 diye etiketlenen deyim, doğrusal enterpolasyon hareket olarak işlem görür. Daha öncede ifade edildiği gibi, dairesel kesici yolu için hareket kademe sayıları tanımlanan tolerans tarafından tespit edilir. Uç nokta koordinatları ile beraber G02 veya G03 kodunun türetilerek dairesel hareketin işlenmesi NC'li tezgahdaki dairesel hareket esnasında kesici aletin konumlandırılmasındaki hassasiyet artık NC işlemcisi tarafından kontrol edildiğini gösterir. Eğer NC kontrol ünitesinin sağlayabileceğinden daha fazla hassasiyet gerekirse dairesel enterpolasyon alt programı kullanılmamalıdır ve CLDATA son işlemci tarafından dairesel hareket olarak işlenmelidir.

6.10 TEZGAH İŞLEMİNİN SONUNU BELİRTEN DEYİM

Bir işleme yönteminin bitirildiği aşağıdaki deyimle gösterilir:

```
END
```

Son işlemci tarafında son işlemcinin tasarlanmasına bağlı olarak M02 veya M30 üretilir.

6.11 TEZGAH KOORDİNAT SİSTEMİNİ TANIMLAYAN DEYİM

Parça geometresinin ve kesici yolunu tanımlamak için kullanılan koordinat sistemi ile takım tezgahının koordinat sisteminin aynı olmadığı veya bir NC

tezgahta işlenecek olan farklı konumlarda iki benzer parçanın olduğu durumlar vardır (Şekil 6-12).

İlk durumda, koordinat sisteminin çevrilmesi gerekir. İkinci durumda, iki parça içinde tezgah koordinat sistemlerinin orijininin doğru ayarlanması şartı ile bu iki parçanın işlenmesi için hazırlanan programlar aynı olabilir. APT programında kullanılan koordinat orijinini tanımlamak için APT'de bir deyim vardır. Bu deyim yapısı

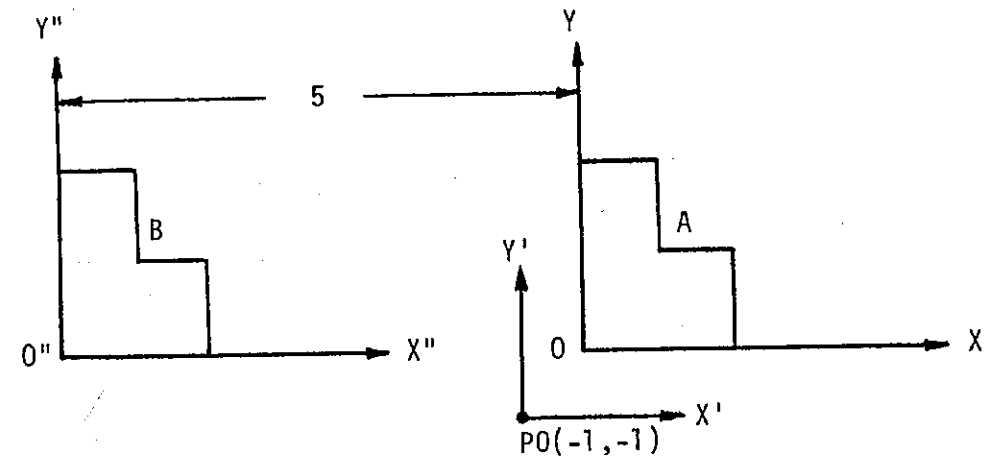
```
ORIGIN/x, y, z
```

burada X, Y ve Z, APT koordinat sistemideki tezgah koordinat sisteminin orijin koordinatlarıdır. Bu APT programcısı tarafından parça geometrisi veya kesici yolunu tanımlamak için uyarlanan koordinat sistemidir. Örneğin, Şekil 6-2'deki APT koordinat sistemi X-Y-Z olsun. A parçası ve kesici yolu APT'de buna göre tanımlanır. Eğer, tezgah koordinat sistem orijini x', y', z' , $PO(-1,-1,0)$ noktasında ise ve A parçasını işlemek için aynı program kullanılacaksa, o zaman orijini 0'dan PO 'a taşımamız gerekir.

Böylece APT koordinat sistemine bağlı olarak tezgah koordinat sisteminin orijini tanımlamak için

```
ORIGIN/-1,-1,0
```

deyimini, kesici yolunu tanımlayan deyimlerden önce belirtmelidir. Daha sonra şekilde de gösterildiği gibi parçanın, 0 noktasında konumlanırken işlenebilir diye



Şekil 6.12 Bir NC'li freze tezgahında işlenen iki benzer iş parçası.

X, Y ve Z yönlerinde 1, ve 0 işleme tabi tutulurken CLDATA değiştirilir. Böylece ORIGIN deyiminin etkisi CLDATA'yı X, Y ve Z yönlerinde -x, -y ve -z ile değiştirmektedir. Aşağıdaki örnek ORIGIN deyiminin başka bir uygulamasını göstermektedir. A parçasını 0 noktasında konumlandırıldığını ve Şekil 6-12'deki parça ve tezgah koordinat sistemlerinin X-Y-Z olarak seçildiğini varsayalım (yani bu iki koordinat sistemi aynı olsun). Eğer A parçası ile aynı fakat tezgah tablasında farklı bir yere, 0" yerleştirilen B parçası A parçası için kullanılan aynı programı ve aynı tezgah koordinat sistemini, X, Y, Z kullanarak işlenecekse o zaman, kesici yolunu 5 inç kaydırmak için bir ORIGIN deyimini tanımlanmalıdır. Böylece, bu iki iş parçasının işlemek için kullanılacak olan program aşağıdaki gibi olabilir.

.....	(A parçasını işlemek için deyimler. Parça ve tezgah koordinat sistemi aynı (yani, X-Y-Z).
ORIGIN/5,0,0	
.....	(A parçası için olan deyimlerin benzerleri. Kesici yolu, orijinal tezgah koordinat sistemi X-Y-Z korunurken B parçası işlenebilsin diye X yönünde -5 inç kaydırılır.

6.12 İŞLEMCİ DENETİM KELİMELERİ

Bu kısımda anlatılan kelimeler, NC işlemcisinin bir özel APT programını işleme yolunu tanımlamak veya değiştirmek için kullanılır. Bunlar. PARTNO, MACHIN, NOPOST, CLPRNT ve FINI.

6.12.1 PARTNO Deyimi

Bu deyim formatı aşağıdaki gibidir:

PARTNO (alfasayısal diziler)

PARTNO kelimesi 1-6 sütunlarında tanımlanmalıdır, bunu takiben 7-72 sütunlarında tanımlanan ve bir yorum veya program etiketi olarak kullanılan bir alfasayısal dizi gelir. Bu deyim bir APT programında ilk deyim olarak tanımlanmalıdır. Bu yorum CLDATA dosyasına yazılır ve son işlemci tarafından yazılmış ve delinmiş çıktıyı etiketlemek için kullanılır. Son işlemcinin tasarımına bağlı olarak bu yorum, delikli şeride insan-okuyabilir bir metin olarak da delinebilir.

Örnek

Bir mil işlemek için hazırlanan APT programının ilk deyimini aşağıda verilmektedir:

PARTNO SHAFT-CUTTING PROGRAM

6.12.2 MACHIN Deyimi

Bu deyim, kullanılacak son işlemciyi ve çıktı yapısını tanımlamak için kullanılır. Hemen PARTNO deyiminden sonra kullanılmalıdır. Bu deyim formatı:

MACHIN/PPname,n1,OPTION,n2,n3,n4

burada:

PPname = kullanılacak son işlemcinin adı

n1 = kullanılan takım tezgahına uyan ve takım tezgahı modülü diye adlandırılan özel işleme alt programı ve son işlemci tarafından kullanılacak bir değişken serisi değerlerini başlangıç değerlerine getiren programı gösteren sayı.

n2 = son işlemci çıktısının şekil ve formatını tanımlayan parametre.

n3,n4 = Kullanıcı için ayrılmış parametreler (bu iki parametre, APT programından son işlemciye bilgi aktarmak için kullanılabilir; bundan dolayı bunlar son işlemciye bağımlıdır).

6.12.3 NOPOST Deyimi

Bu deyim tek bir kelime içerir, NOPOST ve NC işlemcisine bu deyimden sonra son işlemlerin gerekmediği talimatını verir. Yalnız olarak bir satırda tanımlanabilir.

NOPOST

veya virgül ya da noktalı virgül ile birbirinden ayrılan diğer bir deyim ile kullanılır, örneğin:

NOPOST, FEDRAT/2.0

veya

NOPOST; FEDRAT/2.0

6.12.4 CLPRNT Deyimi

NC işlemcisi tarafından bir APT programının işlenmesi, seçilmiş bir son işlemciye girdi dosyası olan, CLDATA diye adlandırılan, ortak bir arabirim kodunu verir. CLDATA aşağıdaki maddeleri içerir:

1. İş parçasını üretmek için program tarafından yönlendirilen ardışık kesici

* Detaylı bilgi için Bölüm 13'de Tablo 13-5'e ve Bölüm 13.4.1.1'e bakın.

2. İşleme özelliklerini tanımlayan son işlemci komutları veya deyimleri,
3. Son işlemciye aktarılması gereken kayıt tipi ve kayıt sıra numarası gibi bilgiler.

Tek bir kelimedenden ibaret olan CLPRNT deyimini tanımlandıktan sonra CLDATA sıralayıcı formatında yazılır. CLDATA 8. ve 11. bölümlerde detaylı olarak açıklanmaktadır.

6.12.5 FINI Deyimi

Bu deyim tek kelime olan FINI'den ibarettir. Parça işleme programını sonunu tanımlar ve programın son kaydı (deyimi) olmalıdır.

6.13 YORUM METNİNİ BELİRLEYEN EK DEYİMLER

PARTNO deyimine ek olarak, bir APT programına (CLDATA dosyasına) ve sonişlemci çıktısına yorum eklemek için daha başka birkaç deyim vardır. Buradaki bunların üçü verilmiştir.

REMARK(alfasayısal dizi)
 PPRINT..... (alfasayısal dizi)
 INSERT..... (alfasayısal dizi)

Eğer gerekirse, 1-6 sütunlarda bu üç kelime tanımlanmalıdır, bunları takiben 7-72 sütunlarda bir alfasayısal metin olarak yorum tanımlanabilir.

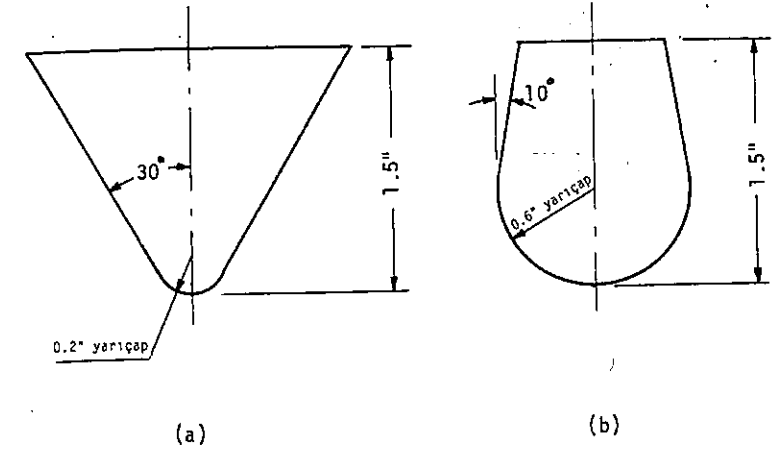
REMARK deyimini parça programına yorum sağlamak için kullanılır. Program listesi dosyasında bu deyim yazılır. Normal olarak, programın içeriğine veya programın özel bir kısmına bazı açıklamalar getirmek için kullanılır. Parça programının işlenmesinde herhangi bir etkisi yoktur.

Benzer şekilde, PPRINT son işlemci kelimesidir. Bunu takip eden yorum son işlemci çıktısı listesi dosyasında yazılır.

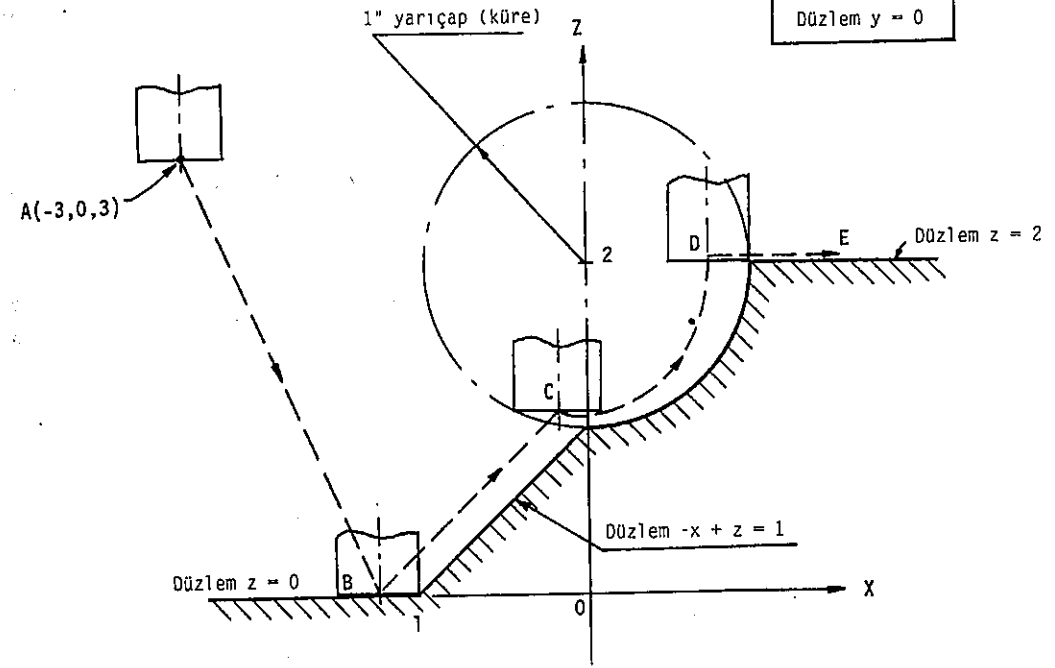
INSERT deyimindeki alfasayısal dizi NC'li tezgah denetimcisine doğrudan girdi olarak, başka bir işleme tabi tutulmaksızın, CLFILE'ına aktarılır daha sonra da son işlemci tarafından kullanılır. Böylece, INSERT deyimindeki bir yorum doğrudan NC çıktısına yazılır ve NC delikli şeride delinir. Bundan dolayı, NC işlemcisi veya son işlemcisi tarafından işlenemeyen bir NC deyimini NC program çıktısına yazmak için INSERT deyimini kullanılabilir.

PROBLEMLER

- 6.1 APT programında bir toleransı tanımlamanın amacı nedir? Bu kesici yolunu veya parça profilini nasıl etkiler? Tolerans ile izin verilen pay arasındaki fark nedir?



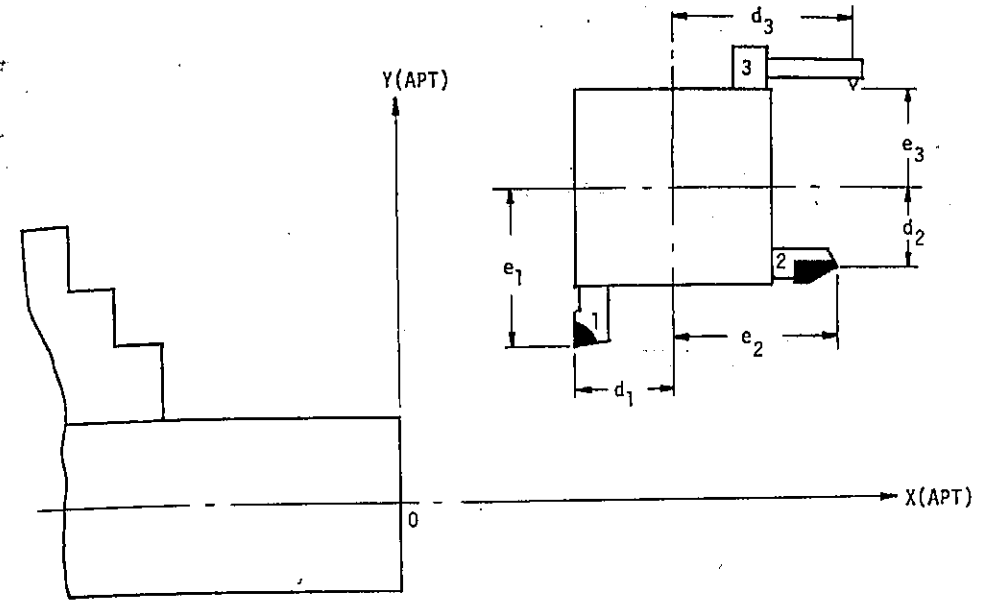
Şekil P6-3



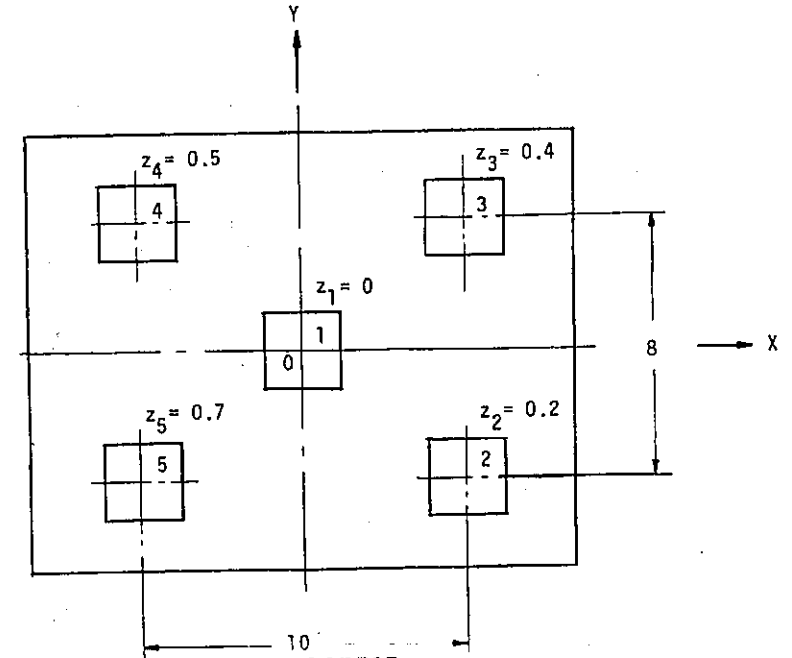
Şekil P6-4

- 6.2 Bir parça profili veya geometrik öge için, SD işlemcisi, tolerans yönlerini (içeri doğru, dışarı doğru) nasıl tespit eder?
- 6.3 Şekil P6-3'te gösterilen kesicileri tanımlayınız.
- 6.4 Şekil 6-4'te gösterilen A-B-C-D-E yolunu tanımlayınız. 0.5 inç çaplı ve 1.5 inç uzunluğunda bir düz uçlu parmak freze kullanılmaktadır.
- 6.5 Aşağıda verilen sırada bir tornalama işlemi gerçekleştirilecektir:
- P0 emniyet noktasına kesiciyi hızlı hareket ettiriniz.
 - Döner başlığı döndürünüz ve 2 no.lu kesiciyi çalışma pozisyonuna getiriniz.
 - Saatın aynı yönünde iş milini 700 devir/dakika'lık bir hızla çalıştırın ve parça profilini yakın olan P1 noktasına kesiciyi hızlı hareket ettiriniz.
 - Soğutma sıvısını açınız ve 0.006 inç/devir'lik ilerleme hızı ile profilin işlemeye başlamak için gerekli pozisyona kesiciyi yavaşça hareket ettiriniz.
 - 0.005 inç/devir'lik bir ilerleme ile profilin ilk parçasını işleyiniz.
 - Devir sayısını 900 devir/dak.ya değiştiriniz.
 - 0.003 inç/devir'lik ilerleme ile profilin ikinci kısmını işleyiniz.
 - Kesiciyi P0 noktasına hızlı olarak hareket ettiriniz.
 - Döner başlığı döndürünüz ve 4 no.lu kesiciyi çalışma pozisyonuna getiriniz.
 - P2 noktasına hızlı olarak gidiniz.
 - 0.002 inç/devir'lik ilerleme ve 500 dev/dak.'lık bir devir sayısı ile üçüncü parça profilini işleyiniz.
 - Kesici aleti P0 noktasına hızlı olarak hareket ettirin, iş milini durdurun ve soğutma suyunu kapatınız. Bu işlemi APT'de tanımlayınız.
- 6.6 Eğer tezgah koordinat sistemi 1 no.lu kesicinin uç noktasının konumuna göre ayarlanmışsa Şekil P6-6'daki 1, 2 ve 3 no.lu kesiciler için LOADTL deyimlerinde tanımlanacak kaydırma değerleri nelerdir?
- 6.7 Şekil P6-7'de gösterilen parçada 1'den 5'e kadar numaralı bölgelerde beş çevre vardır. Bunlar aynı biçimde fakat Z1, Z2, Z3, Z4 ve Z5 gibi yükseklikleri farklıdır. 1 no.lu bölgedeki çevreyi işlemek için yazılan program aracılığı ile 2, 3, 4 ve 5 bölgelerindeki çevreleri işlemek için kesici hareketini yönlendirmek maksadı ile bir APT programı yazınız ve ORIGIN deyimini kullanınız. Kesici aletin başlangıç ve bitiş noktalarının (0, 0,3)'ten başladığını farzedin.
- 6.8 Torna tezgahında çoklu vida açma çevrimi hareketi ve buna karşılık gelen G76 kodu işlenemediğini ve SD işlemci ve son işlemci tarafından türetilmediğini farzediniz. Aşağıdaki SD bloğundaki elde etmek için APT programında vida açma çevrimini nasıl tanımlarsınız?

G76 X 20.752 Z-40.0 K 1.624 D0.5 F3.0A60



Şekil P6-6



Şekil P6-7

Bölüm 7

Geometrik Öğelerin ve Kesici Hareketinin Daha Karmaşık Deyimlerle Tanımlanması

Birçok durumda bir program, döngü programlama, alt programlama ve koordinat sisteminin veya kesici yolunun dönüşümü kullanılarak basitleştirilebilir ya da kısaltılabilir. Bunun için APT Programlama dilinde kullanılan bazı deyimler vardır. Döngü alt programlar ile birlikte matematiksel hesaplama deyimlerinin kullanımı, bir programda hem geometrik öğelerin ve kesici yolunun tanımlanması hem de hesaplanmasını daha kolay kılar. Ayrıca APT, cep frezeleme gibi bazı mutad işleme tanımlamasını basitleştiren deyimleri de içerir. 5. Bölümde anlatılan deyimler tarafından tanımlanması zor ve biçimsiz olan bir çevresel hareket veya çalıştırma işleminin tanımlanması için APT dilinde deyimler bulunmaktadır. Bu bölümde bu konular tartışılmaktadır.

7.1 İNDİSLİ DEĞİŞKENLER VE KONTROL-DALLANMA DEYİMLERİ

Normal olarak indisli değişkenler geometrik öğeler ve alt programlar gibi bir dizi benzer değişkenleri tanımlamak için kullanılır. Bu değişkenler aynı sembolü kullanır ve bir diziyi temsil eder. Örneğin, L(1),....., L1(10) 10 paralel doğruyu veya 10 farklı alt programı temsil eder. Bir programda dizinin herhangi bir elemanı tanımlanmadan önce dizinin boyutu tanımlanmalıdır. Bu aşağıdaki deyimle yapılır.

RESERV/A1,n1,A2,n2,.....

Burada:

A1,A2,.... = farklı diziler için semboller

n1,n2,.... = A1, A2, dizilerinin boyutları (Bunların değerleri 1 ila 32767 aralığında olmalıdır).

RESERV komutu sadece bir kere icra edilebilir. Bundan dolayı bir döngüde kullanılamaz.

İndisli bir değişkenin indisi veya sayıcısı skalar veya matematiksel bir deyim olarak belirtilebilir. Örneğin, eğer I=3 ve J=5 ise A(I+5) indisli değişkeni A(8)'e eşittir ve A(I*J), A(15)'e eşittir. İndis bir tam sayı olmalıdır. Eğer skalar, tam sayı değilse, yuvarlatılır ve sadece tam sayı kısmı kullanılır.

Bazı durumlarda, bir deyimdeki sırası ile listelenmiş indisli değişkenler serisini belirtmeye gerek duyarız (örneğin, PATTERN deyimini). Bu değişkenlerin sayısı fazla olduğunda herbirini tek tek belirtmek sıkıcı olur. Bu problemi çözmek için APT, aşağıda gösterildiği gibi farklı formatların indislerinin tanımlanabilmesini sağlar:

$$A \left(\left\{ \begin{array}{l} a, \text{THRU}, \\ \text{ALL} \end{array} \right\} \left\{ \begin{array}{l} b \\ \text{ALL} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{INCR} \\ \text{DEC} \end{array} \right\}, c \right] \right)$$

Bu, sırası ile listelenmiş A dizisinin bir serisini temsil etmek için bir deyimde belirtilebilir. a ve b skalarları dizideki indisleri gösterir ve c skaları bir indis değişkeninden bir sonrakine geçme indisinin artış veya azalış değeridir. c parametresi ve bundan önce gelen INCR ve DEC düzeltici tanımlanmazsa, bunların SD işlemcisi tarafından INCR, 1 olduğu kabul edilir. Bu üç parametre APT dilinde kabul edilebilir herhangi bir yapıda belirtilebilir. Farklı formatlardaki indisli değişkenlerin anlamları aşağıda anlatılmaktadır.

İndisli değişken

$$A(a, \text{THRU}, b, \text{INCR}, c)$$

Deyimde sırası ile listelenmiş aşağıdaki A dizisinin elemanlarını belirtir.

$$A(a), A(a+c), A(a+2*c), \dots, A(a+n*c)$$

burada:

$$c > 0 \text{ ve } a < b \text{ ise;}$$

$$c < 0 \text{ ve } a > b \text{ ise}$$

$$a+n*c \leq b < a+(n+1)*c,$$

$$a+n*c \geq b > a+(n+1)*c,$$

İndisli değişken

$$A(a, \text{THRU}, b, \text{DECR}, c)$$

şağıdaki deyimde sırası ile listelenmiş A dizisinin elemanlarını belirtir.

$$A(a), A(a-c), A(a-2 \cdot c), \dots, A(a-n \cdot c)$$

burada

$$\begin{aligned} a-n \cdot c \geq b > a-(n+1) \cdot c, & \quad c > 0 \text{ ve } a > b \text{ ise;} \\ a-n \cdot c \leq b < a-(n+1) \cdot c, & \quad c < 0 \text{ ve } a < b \text{ ise} \end{aligned}$$

İndisli değişken

$$A\left(a, \text{THRU}, \text{ALL}, \left\{ \begin{array}{l} \text{INCR} \\ \text{DECR} \end{array} \right\}, c\right)$$

indis artışının işaretine bağlı olarak dizinin ilk ve son indisi ile b parametresinin değiştirilmesi hariç yukarıda gösterilen iki formatla aynıdır. Son olarak, indisli değişken

$$A\left(\text{ALL}, \left\{ \begin{array}{l} \text{INCR} \\ \text{DECR} \end{array} \right\}, c\right)$$

INCR ve DECR düzelticisine ve C parametresinin işaretine bağlı olarak ilk veya son indisle başlayarak ve ilk veya son indisle son bulan A dizisindeki elemanların listesini belirtir. Aşağıdaki örnekler bunların kullanımlarını gösterir.

Örnek 1

Eğer 100 noktayı temsil eden P(1), P(2), ..., P(100) dizisi tanımlanmışsa indisli değişken

$$P(3, \text{THRU}, 25, \text{INCR}, 5)$$

sırası ile listelenmiş P dizisinin aşağıdaki elemanlarını belirtir:

$$P(3), P(8), P(13), P(18), P(23)$$

yukarıda listelenen 5 nokta tarafından tanımlanan bir LCONIC aşağıda verilen herhangi bir deyim tarafından belirtilebilir:

$$\text{LCONIC}/5\text{PT}, P(3, \text{THRU}, 25, \text{INCR}, 5)$$

$$\text{LCONIC}/5\text{PT}, P(3, \text{THRU}, 24, \text{INCR}, 5)$$

$$\text{LCONIC}/5\text{PT}, P(3, \text{THRU}, 23, \text{INCR}, 5)$$

Örnek 2

Yukarıda verilen dizi için farklı yapılarda indisli değişkenlerin anlamları aşağıda gösterilmektedir:

İNDİSLİ DEĞİŞKEN

$$P(75, \text{THRU}, 54, \text{INCR}, -7)$$

$$P(75, \text{THRU}, 54, \text{DECR}, 7)$$

$$P(97, \text{THRU}, \text{ALL})$$

$$P(35, \text{THRU}, \text{ALL}, \text{DECR}, 10)$$

$$P(\text{ALL}, \text{INCR}, -25)$$

BELİRLİ ELEMANLAR

$$P(75), P(68), P(61), P(54) \text{ sıralı listesi}$$

Yukarıdaki ile aynı

$$P(97), P(98), P(99), P(100) \text{ sıralı listesi}$$

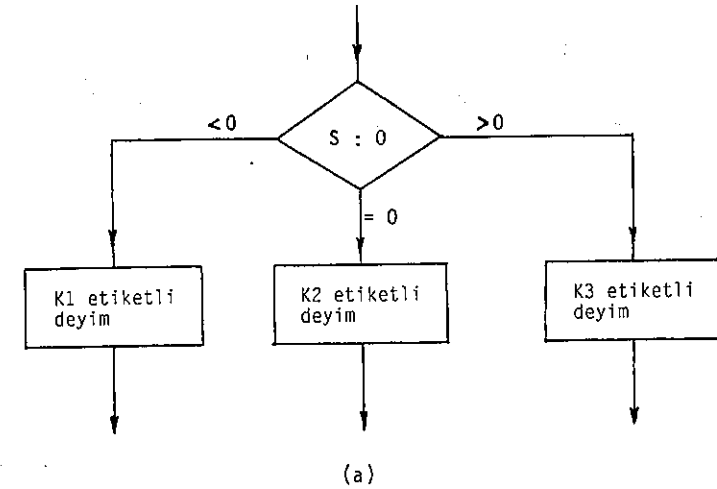
$$P(35), P(25), P(15), P(5) \text{ sıralı listesi}$$

$$P(100), P(75), P(50), P(25) \text{ sıralı listesi}$$

APT dilinde iki deyim bir döngü veya alt program içinde dallanmayı kontrol etmek için kullanılabilir. K1 diye etiketlenmiş bir APT deyimine şartsız dallanmayı (Program akışını) sağlamak için kullanılan deyim

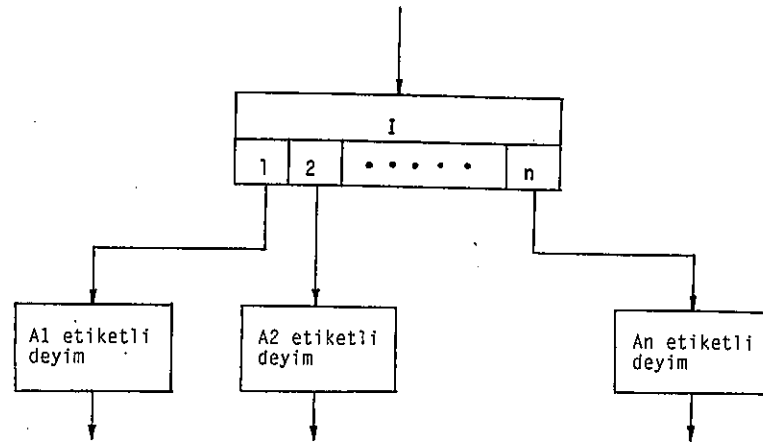
JUMPTO/K1

şartlı dallanma IF ve JUMPTO deyimlerini kullanır. IF deyimini için iki değişik format vardır. Birincisi

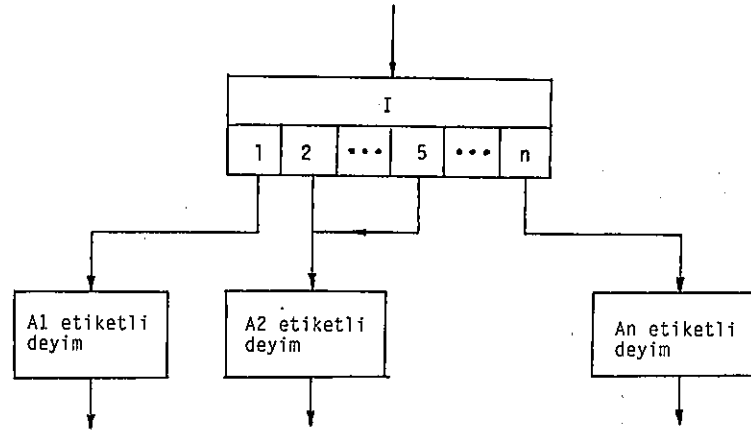


Şekil 7-1 (a) IF(S) K1, K2, K3 deyiminin kontrol akış diyagramı (b) JUMPTO/(A1,A2, ..., An), I deyiminin kontrol akış diyagramı (c) JUMPTO/(A1,A2,A3,A4,A2, ..., An), I deyiminin kontrol akış diyagramı

(devamı var)



(b)



(c)

Şekil 7-1 (devamı)

IF(s) K1,K2,K3

dür ve s 'nin değerine bağlı olarak denetimi yönlendirir. Bu deyimdeki K1, K2 ve K3 sembolleri bir döngüde veya alt programda belirlenmiş üç deyim etiketleridir. Eğer $s < 0$ ise kontrol K1 ile etiketli deyimden gider; eğer $s=0$ ise K2 ile etiketli deyimden gider ve eğer $s < 0$ ise K3 ile etiketli deyimden gider (Şekil 7-1 [a]. s parametresi bir skalar değişken veya bir aritmetik ifade olabilir.

İkinci format

IF(s), APT deyimi

dir. Burada s , "GE" (büyük veya eşittir), "EQ" (eşittir), "GT" (büyüktür), "LT" (küçüktür), ve "LE" (küçüktür veya eşittir) gibi bağıntılı operatörleri içeren bağıntılı bir ifadedir. Bu deyim anlamı: bu bağıntılı ifade s doğru ise bu deyimdeki APT deyimini icra edilir; aksi takdirde bu atlanır ve kontrol bir sonraki deyimden gider. Örneğin, aşağıdaki APT programında

```

.....
.....
.....
IF(A'GE'B), GOTO/PT1
GOTO/PT2
.....
.....

```

} bir döngü veya alt program

eğer $A \geq B$ ise kesici önce PT1 noktasına daha sonrada P2 noktasına gider; aksi takdirde doğrudan PT2 noktasına gider.

Bazen JUMPTO deyimini kontrolü istenilen bir kısma şartlı olarak aktarmak için bir IF deyiminde APT deyimini olarak seçilir. Örneğin, aşağıdaki APT programındaki IF deyimini, A B'den büyük veya eşit olduğu zaman, kontrolü K1 etiketli deyimden aktarır:

```

.....
.....
.....
IF(A'GE'B),JUMPTO/K1
GOTO/PT2
.....
.....
K1)GOTO/PT1
.....
.....

```

} bir döngü veya alt program

Aşağıdaki formata sahip olan diğer bir şartlı dallanma deyimini JUMPTO'dur.

JUMPTO/(A1,A2,.....,An),I

Burada A_1, \dots, A_n bir döngüdeki veya alt programdaki deyimlerin etiketleridir ve I bir tamsayı değişkeni veya sayıcıdır. I değişkeni, örneğin, i değerini varsayarsa denetim, bu deyimdeki listelenen " i "nci deyim olan A_i etiketli deyim'e gönderilir. Bu deyim için kontrol akış diyagramı Şekil 7-1(b)'de verilmektedir. Buradaki sayıcı değerinin $1 \leq I \leq n$ aralığında olması gerektiği açıkça görülebilir. Eğer bu değişken için reel bir sayı kullanılırsa ondalıklı kısım atılır. Böylece bu deyimdeki sayıcı için 1.6 değeri 1 olarak işlem görür. Bu deyimde listelenen deyim etiketleri farklı veya bazıları aynı olabilir (Şekil 711 [c]).

IF deyimini ile karşılaştırıldığında, JUMPTO şartlı dallanma deyimini üç farklı yönden daha fazla kontrolün dallanmasına izin verir. Buna ek olarak, I sayıcısı matematiksel bir ifadeyi kapsayabilir. Örneğin, JUMPTO deyimini ve programdaki sayıcıyı hesaplamak için olan deyim.

.....
.....
.....
.....
.....
I=M+4*tan(C)
JUMPTO/(A1,A2,A3,A4),I

sayıcı olarak iç içe hesaplama ifadeli bir tek JUMPTO deyimini ile değiştirilebilir.

JUMPTO/(A1,A2,A3,A4), (M+4*tan(C))

JUMPTO şartsız dallanma deyimini olduğu için bu deyim bir döngü veya alt program içinde kontrolü yönlendirmek için kullanılır.

7.2 DÖNGÜ PROGRAMLAMA

Bir grup benzer geometrik öğelerin veya bir dizi benzer kesici yolların tanımlanmasına sık sık ihtiyaç duyulabilir. Döngü programlama, bir grup komutları tekrar tekrar icra etmemize izin verir. FORTRAN'da olduğu gibi, bu, DO-döngüsü veya LOOPST ve LOOPND deyimleri ile tanımlanan bir döngü içinde kontrolü aktarma yolu ile gerçekleştirilebilir. Döngü, seçenekler arasında bir karar vermek için de kullanılabilir.

7.2.1 DO-Döngüsü

DO döngüsünün biçimi ve yapısı Tablo 7-1'de gösterilmektedir. İlk deyim, DO/....., ve K diye etiketlenen son deyim bir DO döngüsünü tanımlar. Bunlar verilen sayı kadar döngü içindeki deyimleri tekrar tekrar icra ettirirler. DO döngüsü girildiği zaman, I sayıcısı n_1 değerini alır. Her tekrarlama sonra bunun değeri n_3 kadar artırılır. Bu tekrarlama, I n_2 'den küçük veya n_2 'ye eşit olana kadar devam eder. I , n_2 'den büyük olduğu zaman, K ile etiketli olan deyimden sonra gelen deyim icra edilir. DO döngüsünün akış diyagramı da Tablo 7-1'e eklenmiştir.

TABLO 7-1 DO-DÖNGÜSÜNÜN YAPISI VE AKIŞ DİYAGRAMI

Format	Akış diyagramı
DO/K,I=n1,n2,n3 K)(APT deyimini)	
Not: <ol style="list-style-type: none"> $n_3=1$ ise, bu atlanabilir. n_1, n_2 ve n_3 s kadar veya iç içe aritmetik ifadeler olabilir. n_3 pozitif veya negatif olabilir. Eğer $n_3 > 0$ ise n_2, n_1'den büyük olmalıdır; eğer $n_3 < 0$ ise, n_1, n_2'den büyük olmalıdır. 	
Örnek: Aşağıdaki DO döngüsü 20 noktayı tanımlar:	
DO/K,I=1,10 P1(I)=POINT/I,I,0 J=I+15 P2(I)=POINT,J,I,0 K)CONTIN	

İcra edilebilir kukla bir deyim olan CONTIN, bazen DO döngüsünün son deyimini olarak kullanılır (Tablo 7-1'deki K etiketli olan), bunun fonksiyonu FORTRAN'daki CONTINUE deyimini ile aynıdır (yani bu bilgisayar programının icrasına devam etmesi için yönlendirir).

Döngüdeki son deyimden fonksiyonlarından biri, kontrolü döngünün başına

geri göndermektir. Bu sebeple, kontrolü diğer deyimlere gönderen veya yeni bir döngü veya alt programı tanımlayan deyimler son deyim olarak kullanılamazlar. Bununla beraber, DO döngüsünün son deyimini şunlardan herhangi birisi olabilir: CALL, IF, JUMPTO, FINI, MACRO, TERMAC veya bir başka DO deyimini. DO döngüsünün ana gövdesinde olan deyimler tekrar tekrar T kere icra edilir. $T = \text{INT}(\text{TGF}((n2-n1)/n3)+1)$ 'dir (Tablo 7-1'e bakınız). Bununla beraber şartlar, DO döngüsünün aralığı dışına kontrolü aktarmak için ayarlanabilir ve böylece DO döngüsü erkenden durdurulabilir. DO döngüsü içinde JUMPTO deyimini ile birlikte bir IF deyimini, kontrolü aktarmak için kullanılabilir. Örneğin, aşağıdaki DO döngüsünün çalışmasını IF deyimini durduracaktır; daha sonra kontrol, DO döngüsü dışında, fakat DO döngüsünü içine alan döngünün içindeki K2 ile etiketli deyime gider:

```

.....
LOOPST
.....
DO/K1,I=1,S, $$$THE VALUE OF S MUST HAVE BEEN ASSIGNED BY A PREVIOUS
                $$$STATEMENT
.....
IF((I-R1)'GT'25),JUMPTO/K2
.....
K1) CONTIN
.....
K2).....
.....
LOOPND
.....

```

K2 etiketli deyim döngü içinde değilse veya DO döngüsünü içine alan bir alt programda ise kontrolün DO döngüsü dışına aktarılması başarısızlıkla sonuçlanır.

Bir DO döngüsü ilk DO deyimini hariç herhangi bir noktada girilemez. DO döngüleri iç içe yazılabilir; bu iç içe yazılımın en büyük değeri 31'dir. Fakat, bunlar birbirini üzerine binmemelidir. Çünkü bu mantıksal hatalar doğurur. Örneğin, aşağıdaki program parçası

```

DOIK,I=n1,n2,n3
.....
DO/K1,J=n4,n5,n6
.....
.....

```

```

K1)CONTIN
.....
.....
K)CONTIN

```

İki DO döngüsüne sahiptir, birisi diğerinin içine yerleştirilmiştir. İlk DO döngüsü bilgisayarı, içindeki deyimleri $\text{INTGF}((n2-n1)/n3)+1$ kere tekrarlamak için yönlendirir. Her seferinde DO döngüsü icra edilir ve içteki DO döngüsü $\text{INTGF}((n5-n4)/n6)+1$ kere icra edilir. Diğer yandan, aşağıdaki program parçası

```

DOIK,I=n1,n2,n3
.....
DO/K1,I=n4,n5,n6
.....
K)CONTIN
.....
K1)CONTIN

```

birbiri üstüne binen iki DO döngüsü içerir. İlk DO döngüsü, bilgisayarı döngü içindeki deyimleri $\text{INTGF}((n2-n1)/n3)+1$ kere icra etmesi için yönlendirir. İkinci döngünün ilk deyimini icra edildiğinde bu iki DO döngüsünün aralığı dışında kalan deyimleri de içeren ikinci döngü içindeki deyimleri $\text{INTGF}((n5-n4)/n6)+1$ kere icra etmek için bilgisayara komut verir. Bu iki DO döngüsü tarafından verilen komutlar birbirini çelişkiye girer. Sonuç olarak bu program icra edilemez.

Bir DO döngüsü bir alt programda (MACRO) veya LOOPST ve LOOPND deyimleri ile tanımlanan döngüde kullanılabilir. DO döngüsü dışında tanımlanan bir alt programı çağırmak gibi kontrolü DO döngüsünün aralığının dışına aktarmak için olan deyimlerde DO döngüsünün içine konabilir. Bu durumda, alt program icra edildikten sonra kontrol, DO döngüsündeki alt programı çağıran deyimden sonraki deyime geri döner.

Bir alt program (MACRO) tekrar tekrar tanımlanamaz; bundan dolayı MACRO deyimini DO döngüsü içine konmamalıdır.

DO döngüsünün kullanımını açıklayan iki örnek aşağıda verilmektedir.

Örnek 1

Şekil 7-2(a)'da üstte yatay dairesel ve tabanda elitik kesitli olan yarım koni şekilli bir parça gösterilmektedir. $z=0$ ve $z=3$ arasında herhangi bir z seviyesinde bu parçanın yatay kesiti şu formülle tanımlanır

$$x^2/(3-z/1.5)^2 + y^2/(2.5-z/3)^2 = 1$$

Bu formülden de görüleceği üzere, 2 inç yarıçaplı bir daire olan üst yüzey hariç diğer bütün ara kesitler elipstir. Birbiri ile eşit aralıklı (0,15 inç) üst ve taban yüzeylerini de içeren 11 yatay arakesiti tanımlamak için kullanılan DO döngüsü aşağıdaki gibi olabilir:

RESERV/CROS,11 \$\$ CROS(I) YATAY KESİTİN SEMBOLÜDÜR

DO/A1,I=1, 11

Z=1.5-0,15(I-1)

IF(I'GT'1),JUMPTO/A2

CROS(I)=CIRCLE/0,0,1,5,2

JUMPTO/A1

A2) CROS(I)=ELLIPS/CENTER,(0,0,Z1),(3-Z1/1.5),(2.5-Z1/3),0

A1) CONTIN

Örnek 2

Şekil 7-2(b)'de gösterilen desenden sonra bir düzlem yüzey, X-Y düzlemindeki kesici yolu ile üç pasoda bir alın freze çakısı ile işlenecektir. İlk iki pasodaki talaş derinliği 0,2 inçtir ve sonuncusunda 0,05 inçtir. Başlama ve bitiş noktalarının (-1,3,1,1) olduğunu farzediniz. Kesici hareketini tanımlayan DO döngüsü aşağıdaki gibi olabilir:

FROM/-1,3,1,1

GODLTA/-0,45

DO/A4,I=1,3

JUMPTO/(A1,A1,A2),I

A1) GODLTA/-0,3

JUMPTO/A3

A2) GODLTA/-0,15

A3) GODLTA/6,6,0,0

GODLTA/0,2,0

GODLTA/-6,6,0,0

GODLTA/0,2,0

GODLTA/6,6,0,0

GODLTA/0,1

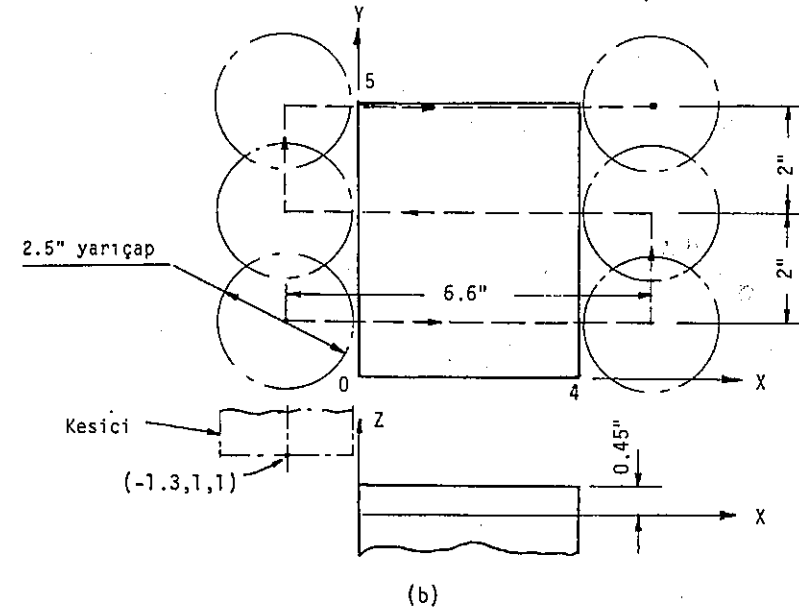
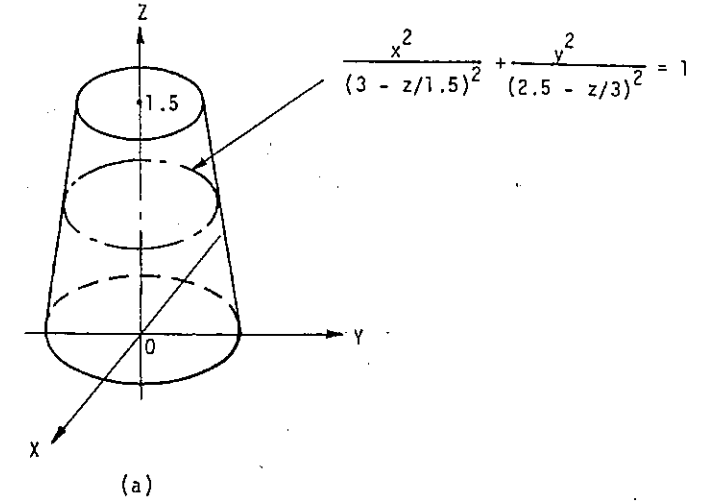
GODLTA/-6,6,-4,0

A4) CONTIN

GOTO/-1,3,1,1

7.2.2 LOOPST ve LOOPND Deyimleri Tarafından Tanımlanan Döngü

Bir döngü, döngü başlama deyimi, LOOPST ve döngü sonu deyimi, LOOPND ile



Şekil 7-2 (a) Bir eliptik fonksiyonla tanımlanmış yatay kesitli yarım koni şekilli parça. (b) Bir alın frezelenme işlemi için kesici yolu.

de tanımlanabilir (Tablo 7-2). Bu iki deyim arasında döngünün ana gövdesi yer alır. DO döngüsünde olduğu gibi sayıcıyı saymak için ve kontrolü yönlendirmek amacıyla bir DO deyimi olmadığı için döngü sayıcısı ek deyimlerle tanımlanmalı-

dır. Bir LOOPST-LOOPND döngüsü sadece başlama deyimi ile (yani LOOPST) çalıştırılabilir. Bununla beraber programda sadece birkere icra edilebilir. IF ve JUMPTO deyimleri LOOPST-LOOPND döngüsü içinde kontrolü yönlendirmek için sık sık kullanılır. Tablo 7-2 bir döngü içinde APT deyimlerinin (N kere) tekrarlanan icralarını anlamak için iki yol göstermektedir. İlk algoritmada, döngü şartlı bir dallanma deyimi tarafından kontrol edilir.

IF(I-N)A1,A1,A2

ve ikincisinde şartlı IF deyimi ile şartsız dallanma deyimi, JUMPTO tarafından kontrol edilir. Kontrolün LOOPND'den başka bir deyimden döngü dışına aktarılmasına izin verilmez.

LOOPST-LOOPND'de kullanılan deyimlerin etiketleri sadece döngü içinde etkilidir. Bunlar başka bir döngüde veya bir alt programda (MACRO) kullanılabilir. LOOPST-LOOPND döngüsü içinde kullanılmayan APT deyimleri FINI, MACRO ve RESERV gibi tekrar tekrar icra edilemeyen deyimlerdir.

DO döngüsü ile karşılaştırıldığında, LOOPST-LOOPND döngüsünün kullanım alanlarından biri, tekrar sayısı tahmininin zor olduğu zaman bir dizi işlemlerin tekrar edilmesidir. Örneğin, Şekil 7-3(a)'da gösterilen eğik yüzeyin 3 eksenli bir freze tezgahında işleneceğini farzedelim. Kesici şekilde gösterilen zigzag yolda hareket ediyor. İzin verilen dalgalanma yüksekliği 0.005 inç'tir. Farklı seviyedeki kesici hareketi için kullanılan tahrik yüzeyleri olan dikey düzlemlerin $PL(i)$, $i = 1, 2, \dots$, tanımlanması için birbirini takip eden seviyelerde kesicinin eksen doğrusu konumları $x(i)$ izin verilen dalgalanma yüksekliğine, H, bağlı olarak hesaplanabilir. Kesici alet B yüzeyine dokunana kadar hesaplama tekrarlanmalıdır. Bu hesaplama işlemini gerçekleştirmek ve $PL(i)$, $i=1, 2, \dots$ düzlemlerini tanımlamak için bir LOOPST-LOOPND döngüsü tasarlanabilir. Bu döngü aşağıdaki gibi atabilir:

PHI=ATANF(1.5)

R=3/8

H=0.005

X(1)=R*COSF(90-PHI)

D1=2*SQRTF(2*R*H-H*H)

D=D1*COSF(PHI)

X1=1+R*TANF(PHI/2)

PL(1)=PLANE/1,0,0,X(1)

LOOPST

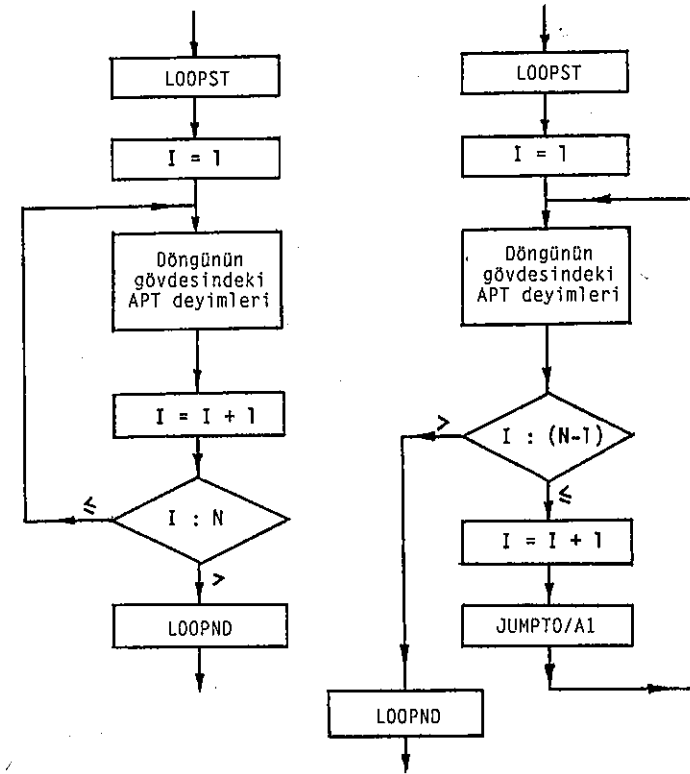
I=2

A0) X(I)=X(I-1)+D

IF(X(I)'GE'X1),JUMPTO/A1

TABLO 7-2 LOOPST-LOOPND DÖNGÜSÜNÜN FORMATI VE BİR DÖNGÜDE KONTROLÜ GERÇEKLEŞTİRMEK İÇİN İKİ YOL

Format	Örnek 1	Örnek 2
LOOPST	LOOPST	LOOPST
(döngünün gövdesi)	I=1	I=1
	A1).....	A1).....
	(APT deyimleri)	(APT deyimleri)
LOOPND	I=+1	IF(I-N-1)B1,B1,B2
	IF(I-N)A1,A1,A2	B1)I=+1
	A2)LOOPND	JUMPTO/A1
		B2)LOOPND



```

PL(I)=PLANE/1,0,0,X(I)
I=I+1
JUMPTO/A0
A1) X(I)=X1
    PL(I)=PLANE/1,0,0,X1
LOOPND

```

LOOPST-LOOPND döngüsü karmaşık durumlarda karar vermek için de kullanılabilir. Örneğin, Şekil 7-3(b)'de kesici L1, L2, L3 ve L4 doğrularından oluşan profili işleyecektir. L1, L2 ve L3 doğrularının oluşturduğu köşede, kesici sola doğru hareket ettiği zaman iki doğru (L2 ve L3) hangisinin kesiciye temas edeceğini tahmin etmek zordur. LOOPST-LOOPND döngüsü, doğru denetim yüzeyinin seçimini sağlamak ve doğru kesici hareketini tanımlamak için kullanılabilir. * Döngü aşağıdaki gibi görülebilir:

```

.....
L1=LINE/XAXIS
L2=LINE/0,0,ATANGL,95
L3=LINE/(-1.2*TANF(5)),1.2,ATANGL,70
L11=LINE/PARLEL,L1,YLARGE,1
L21=LINE/PARLEL,L2,XLARGE,1
L31=LINE/PARLEL,L3,XLARGE,1
P1=POINT/INTOF,L31,L11; OBTAIN,POINT/P1,X1,Y1,Z1
P2=POINT/INTOF,L21,L11; OBTAIN,POINT/P2,X2,Y2,Z2
.....
.....

```

(THE TOOL HAS MOVED TO POINT A)

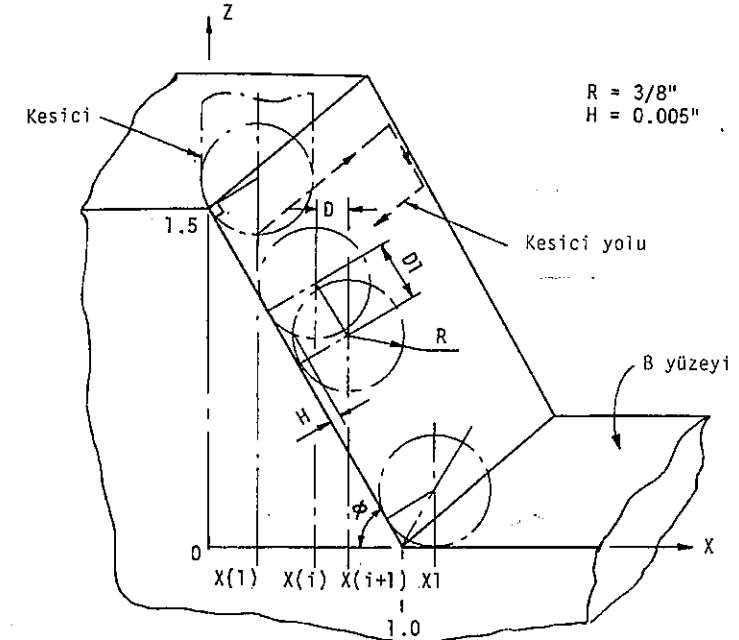
```

LOOPST
INDIRV/-1,0,0
IF(X1-X2)A1,A1,A2
A1) TLRGT,GOFWD/L1,L3
    JUMPTO/A3
A2) TLRGT,GOFWD/L1,L2
    GORGT/L2,L3
A3) GORGT/L3,PAST,L4
LOOPND

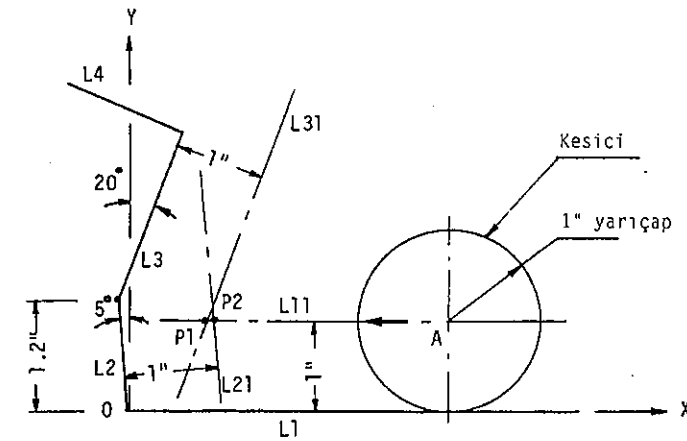
```

Bir alt program, kesici yarıçapı ve köşe profili çeşitlerini gözönünde tutmayı genelleştirebilen karar verme işlemi üstünlüğüyle, LOOPST-LOOPND döngüsü fonksiyonunu gerçekleştirme durumunda da kullanılabilir.

*Bu örnek sadece döngü programlamanın kullanımını açıklamak için burada verilmiştir. Gerçekte, APT dili bu problemi çözmek için çoklu kontrol yüzeyi deyimini içerir. Detay için Bölüm 7.8.2'e bakınız.



(a)



(b)

Şekil 7-3 (a) Üç eksenli NC'li freze tezgahında açılı bir yüzey işlenecektir. (b) L1-L2-L3-L4 parça profili ve freze kesicisi. İki yüzeyden, L2 veya L3 hangisinin kesici ile önce temas edeceğini tespit etmek için hesaplama gerekir.

7.3 ALT PROGRAMLAR (MACRO) İLE PROGRAMLAMA

APT'de bir alt program, bilgisayar hafızasında okunan ve saklanan ve bir çağırma deyimi ile çağrılana kadar icra edilmeyen, iş parçasını işleme programının bir parçasıdır. Bir alt program, MACRO deyimi ile başlamalı ve TERMAC deyimi ile son bulmalıdır. Bu iki deyim arasında, çağrıldığı anda icra edilecek bir dizi komut tanımlanır. Bir alt program veya MACRO alt programa referans olabilen ve CALL deyiminin kullanımı ile icra edilebilen bir isme veya sembole sahip olmalıdır.

MACRO deyiminin formatı şöyledir

sembol=MACRO/M1,M2,.....,Mn

burada

sembol = aşağıdaki sınırlamaları içeren alfasayısal karakterlerin herhangi bir birleşimi:

1. En fazla karakter ve rakam sayısı 6'yı geçmemelidir.
2. İlk karakter bir harf olmalıdır.

M1,M2,.....,Mn= MACRO çağrıldığında, alt programda kullanılan ve herhangi bir değer veya kelimelerin atanabildiği program veya MACRO değişkenleri. MACRO değişken: alt programda kullanılmazsa, deyim basitleştirilmiş olur:

sembol=MACRO

Alt programın veya MACRO'nun son deyimi

TERMAC

olmalıdır. Bu MACRO'nun son bulunduğunu gösterir. CALL deyimini ile çağrılabilen MACRO'nun formatı

CALL/sembol,M1=b1,M2=b2,.....,Mn=bn

dır. Buradaki "sembol" elemanı çağrılacak MACRO'nun ismidir ve b1-bn, M1'den Mn'e kadarki MACRO değişkenlerine atanan değerler veya kelimelerdir. Aşağıda ilerleme hızı, soğutucu tipi ve kesici hareketini tanımlamak veya değiştirmek için alt program kullanan bir APT program örneği verilmektedir:

.....

 M1=MACRO/A,B,C
 FEDRAT/A

COOLNT/B
 GODLTA/0,0,C
 TERMAC

.....

 CALL/M1,A1=2.0,B=ON,C=1.0

Alt programın veya MACRO'nun CALL deyiminden önce tanımlanması gerektiğine dikkat edilmelidir. Bilgisayar CALL deyimine ulaşıncaya kadar alt programı icra etmez. Aşağıdaki kelimeler MACRO değişkenine atanamazlar: IF, JUMPTO, CALL, MACRO, TERMAC, FINI, LOOPST ve LOOPND.

MACRO programlama ile ilgili diğer kurallar şunlardır:

1. Bir alt program sadece bir kere tanımlanabilir. Böylece, bir döngü içinde veya bir diğer alt programda tanımlanamaz.
2. MACRO'ların mümkün olduğu kadar küçük tutulması tavsiye edilir. MACRO dışında tanımlanabilen deyimler alt programın içine konmamalıdır, böylece gereksiz tekrar işlemlerden kaçınılmış olur.
3. LOOPST-LOOPND döngüsü bir MACRO içinde tanımlanamaz. Ayrıca MACRO, FINI deyimini içermemelidir.
4. Bir MACRO, diğer bir MACRO'dan belirlenmiş CALL deyimini ile çağrılıp kullanılabilir.
5. JUMPTO deyiminde bir deyim etiketini temsil etmek için bir MACRO değişkeninin kullanımına izin verilmez.

Genellikle, bir alt program, birden fazla icra edilecek bir dizi komutları tanımlamak için kullanılır. Ayrıca, sık sık bu komutlardaki belirli parametrelerin değerlerini, MACRO'nun her çağrılmasında işlem ihtiyacına göre değiştirmek gerekir. Örneğin, aşağıdaki program parçası iki grup çember tanımlar:

.....
 A1=MACRO/R1,R2,N1,N2
 DO/B1,I=N1,N2
 IF(I GT 5)JUMPTO/B2
 A=72*(I-1)
 JUMPTO/B3
 B2) A=90*(I-6)
 B3) C(I)=CIRCLE/(R1*COSF(A)),(R1*SINF(A),R2
 B1)CONTIN
 TERMAC
 CALL/A1,R1=2,N1=1,N2=5,R2=0.5 \$\$\$DEFINING THE FIRST SET OF CIRCLES

```

$$ C(1 THRU 5) EQUALLY DISTRIBUTED ON THE CIRCLE OF 2" RADIUS
CALL/A1,R1=3.0,N1=6,N2=9,R2=0.25 $$DEFINING THE 2ND SET OF CIRCLES
$$ C(6 THRU 9) EQUALLY DISTRIBUTED ON THE CIRCLE OF 2.5" RADIUS
.....

```

İlk grup 0.5 inç yarıçaplı beş çemberden ibarettir. Bunlar 2 inç yarıçaplı ve merkezi (0,0) da olan bir çember üzerinde eşit olarak dağıtılmıştır. Aynı alt program yarıçapı 3 inç ve merkez (0,0) da olan diğer bir çember üzerinde eşit olarak dağıtılan 0.25 inç yarıçaplı dört çemberden oluşan ikinci grup çemberin tanımlanması için de kullanılır. Böylece parametreler veya MACRO değişkenleri, farklı çember gruplarına karşılık gelen değerlere atanırlar.

Bir alt program çağrıldığında sadece bir kere icra edilebilir. Fakat, eğer çağırma deyimi bir döngü içinde ise ve birkaç kere icare edilirse alt programda bu tekrar sayısı kadar çağrılabilir. Örneğin, iki grup çemberin tanımlanması için kullanılan yukarıda listelenmiş program parçası aşağıdaki program ile de yer değiştirebilir:

```

.....
A1=MACRO
C(I)=CIRCLE/(R1*COSF(A)),(R1*SINF(A)),R2
TERMAC
DO/B1,I=1,5
  A=72*(I-1)
  CALL/A1,R1=2,R2=0.5
B1)CONTIN
D)/B2,I=6,10
  A=90*(I-6)
  CALL/A1,R1=2.5,R2=0.25
B2) CONTIN
TERMAC
.....

```

Bu örnekte, bir alt programa gerek yoktur. Çünkü DO döngüsüne konabilen alt program sadece bir deyimden ibarettir. Fakat geometrik tanımlama deyimlerinin birkaçı bulunduğu, bir alt programa gerek duyulacağı açıktır. Genellikle, döngülerin ve alt programların mantıksal birleşimleri, karışık tekrarlı hesaplama ve karar verme işlemini gerektiren problemleri çözebilir.

7.4 DÖNGÜ VE ALT PROGRAMLARDAKİ YAYGIN PROGRAMLAMA HATALARI

Mantık hatalarından başka, bu kısımda tartışılan, yeni başlayanlar tarafından sık sık yapılan bazı hatalar vardır.

APT programlamanın genel kurallarından birisi, bir APT programında bir geometrik öğeyi temsil eden sembol diğer bir geometrik öğe olarak veya hatta kendisi ile aynı geometrik öğe olarak tekrar tanımlanamaz. Örneğin, aşağıdaki programda

```

.....
P1=POINT/0,0,0
.....

```

```

.....
P1=POINT/0,1,0
.....

```

(0,0,0) noktasını temsil eden P1 sembolü, (0,1,0) olarak tekrar tanımlanmış. Bu durumda 1321* numaralı bir hata mesajı verilecektir. Bir döngüde veya alt programda geometrik tanımlama deyimleri belirtilirse döngü icra edildiğinde veya alt program tekrar tekrar çağrıldığında geometrik öğeler için olan semboller tekrar belirlenir. Böylece aynı hataya (No. 1321) götürür.

Bu problemi çözmek için iki yol vardır. Birincisi mümkün veya uygunsa bu geometrik tanımlama deyimleri döngü veya MACRO dışında belirlemektir. Alternatif olarak bu semboller dizi isimleri olarak kullanılabilir ve bir geometrik öğe tanımlandıkça indiste değiştirilebilir. Eğer programlama geometrik öğeler için sembollerin tanımlanmasını gerektirirse geometrik tanımlama deyimlerinden önce CANON/ON deyimi belirlenebilir. Bu deyim geometrik öğe sembollerinin sonraki deyimlerde tekrar tanımlanmasına izin verir. CANON/OFF deyimi, bir geometrik öğeye bir kez atama yapıldığında tekrar tanımlanmasına izin vermeyen bir semboldür ve APT-AC-SD işlemcisinin varsayılan modunu temsil eder.

Bir döngü veya MACRO'da çevresel hareket deyimleri tanımlandığı zaman bunlar genellikle tekrar tekrar icra edilir. Bu gibi durumda, her tekrarlanmanın başında kesicinin doğru başlama konumuna yerleştirildiğinden emin olmak için kesici başlama konumunun kontrol edilmesi gerekir. Bu aşağıdaki örnekle gösterilebilir. Şekil 7-4 iç bir profili işlemek için bir parmak frezeyi yönlendiren bir APT programını göstermektedir. Kaba işleme esnasında tahrik ve denetim yüzeyleri üzerinde 0.02 inçlik bir izin verilen pay (işleme payı) aşağıdaki deyim tanımlanmasıyla bırakılmıştır.

```

THICK/0,0.02,0.02

```

* Hata mesajları için Bölüm 8.7'ye bakınız.

Bitirme işleminde, izin verilen pay aşağıdaki deyim ile kaldırılır:

THICK/0,0,0

Fakat, işlemeyi bitirdikten sonra, kesici A1 etiketli deyimde belirli tahrik yüzeyinden 0.02 inç uzaktadır. Bu kesicinin doğru başlama konumuna yerleştirilmediğini ve çevresel harekete başlamak için hazır olmadığı anlamına gelir. Bundan dolayı, APT-AC SD işlemcisi, kesicinin hareket dizisinin başlangıcında tahrik yüzeyine göre toleransın dışında olduğunu gösteren 2209 nolu hata mesajını verir. Bu probleme çözüm için,

GO/C1,PL1,L41

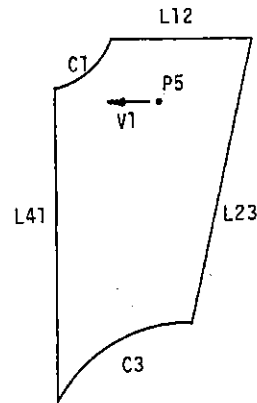
başlama hareket deyimini B1 ve B2 diye etiketlenmiş deyimler arasına sokulabilir. O zaman, kesici çevresel harekete başlamak için gerekli konuma yerleştirilecektir.

GO/C1,PL1,L41

```

.....
INNER=MACRO
A1)TLLFT,GOLFT/L41,C3
ARCSLP/ON
GOLFT/C3,L23
GOLFT/L23,L12
GOLFT/L12,C1
ARCSLP/ON
GOLFT/C1,L41
TERMAC
.....
FROM/P5
THICK/0,0.02,0.02
INDIRV/V1
GO/C1,PL1,L41
CALL/INNER
B1)THICK/0,0,0
B2)CALL/INNER
.....

```



Şekil 7-4 Bir parçanın iç profilini işlemek için bir APT programı. PL1 parça yüzeyi (gösterilmeyen) kesici eksenine diktir. Bu programda bitirme kesici yolu yanlış olarak tanımlanmıştır, çünkü düzgün bir başlama konumu yoktur.

7.5 GEOMETRİK ÖĞELERİ VE KESİCİ YOLUNU TANIMLAMAK İÇİN MATEMATİK HESAPLAMA DEYİMLERİNİN DÖNGÜLER VE/VEYA ALT PROGRAMLARLA KULLANILMASI

NC koddaki programlarıyla karşılaştırıldığında, APT'de programlama, programdaki matematiksel hesaplama izin verme üstünlüğü de vardır. APT'de çok çeşitli geometrik öğelere izin verilmesine rağmen, tanımlanamayan birçok öğe daha vardır. APT'de izin verilmeyenler için, önce farklı noktaların koordinatlarını onların ana hatlarında hesaplayabiliriz ve sonra onları LCONIC veya çizelgelenmiş silindir olarak tanımlayabiliriz. Genellikle, profil ve eğim devamlılığının korunabilmesi için, bu hesaplamalar bilgisayar tarafından yapılmalıdır. Kesici yollarının bir diğerine benzediği fakat, örneğin, kesme derinliğinde ve profilin boyutlarında farklılıkların olduğu durumlar da vardır. Alışlagelmiş matematiksel hesaplamalar böyle değişikliklerin tanımlanmasını döngülerde belirtebilir.

4. Bölümde, bir dış teker eğrisi (Wankel motor hücresinin profili) tanımlamak için matematiksel hesaplama deyimlerinin kullanımı bir örnek aracılığı ile gösterilmektedir. Profili tanımlamak için 10 derece kadar artışı bir açı için bile yaklaşık 100 deyim gerek duyulduğu tespit edilmiştir. Aşağıda, programın büyüklüğünü büyük ölçüde azaltan döngüyle veya MACRO ile matematiksel hesaplama deyimlerinin birlikte kullanımını göstermek için iki örnek verilmiştir.

Örnek 1

4. Bölümün 4.13 kısmında üst üste binmiş dokuz koni şeklindeki Wankel motor hücresinin profili yaklaşık 100 deyim ile tanımlanmaktadır. Bu program parçası döngü programlama aracılığı ile aşağıdaki gibi sadeleştirilebilir:

```

.....
RESERV/X,36,Y,36,E,9,P,36
.....
DO/A1,I=1,36
  A=10*I
  B=3*A
  X(I)=60*COSF(A)+10*COSF(B)
  Y(I)=60*SINF(A)+10*SINF(B)
A1) P(I)=POINT/X(I),Y(I)
DO/A2,J=1,8
A2) E(J)=LCONIC/5PT,P(J*4-3),P(J*4-2),P(J*4-1),P(J*4),P(J*4+1)
E(9)=LCONIC/5PT,P(33),P(34),P(35),P(36),P(1)
.....

```

Sadece 10 deyimden oluşan programın bu parçası, kısım 4.13'te (Bölüm 4) gösterilen programdaki 117 deyimine yerine geçmektedir.

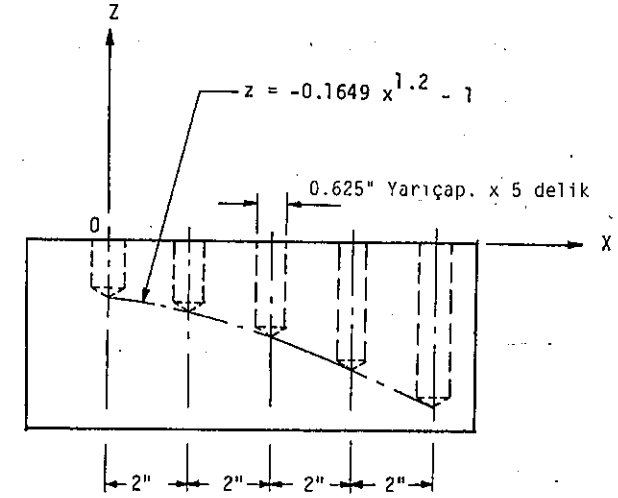
Örnek 2

Şekil 7-5, X ekseninde yerleştirilen farklı derinlikteki beş delikli bir parçayı göstermektedir. Bu beş deliği delmek için bir SD'li matkap tezgahını yönlendiren program aşağıdaki gibi olabilir.

```

.....
D1=MACRO/Z1
RAPID;GODLTA/2.0,0,0
GODLTA/-Z1
RAPID;GODLTA/Z1
TERMAC
.....
FROM/-5.0,4.0,3.0 $$$ THE STARTING POSITION IS (-5,4,3).
RAPID;GOTO/(-2,0,2.1)
FEDRAT/2.0
DO/A1,I=1,5
  Z1=0.1649*X**1.2+1
  CALL/D1
A1) CONTIN
RAPID;GOTO/-5,4,3
.....
Yukarıda gösterilen program aşağıdaki gibi de yazılabilir
.....
FROM/-5.0,4.0,3.0
RAPID;GOTO/(-2,0,2.1)
FEDRAT/2.0
DO/A1,I=1,5
  Z1=0.1649*X**1.2+1
  RAPID;GODLTA/2.0,0,0
  GODLTA/-Z1
  RAPID;GODLTA/Z1
A1) CONTIN
RAPID;GOTO/-5,4,3
.....

```



Şekil 7-5 Farklı derinlikteki beş delikli bir parça

7.6 DÖNÜŞÜM MATRİSLERİNİ TANIMLAYAN DEYİMLER

7.6.1 Bir Dönüşümün Matrisle Gösterimi

Genellikle, verilen bir uzaysal konum ve verilen uzaysal yönlendirmedeki bir geometrik öğe, hem çeviri (paralel kayma) hem de dönme ile herhangi bir uzaysal konumda ve uzaysal yönlendirmede yerleştirilebilir. Çeviri deyimini ile, bir geometrik öğenin, kesici yolunun, veya koordinat sisteminin bir konum veya yönlendirmeden diğerine hareketini kastediyoruz. Çeviri ve dönme, değişimin iki temel tipidir. Bir geometrik öğenin değişimi, çeviri, dönme veya her ikisini de içerebilir. Bir nokta, herhangi bir konuma çeviri ile yerleştirilebilir. Fakat, genellikle noktadan başka bir geometrik öğeyi istenen uzaysal konuma ve yöne getirmek için iki tip değişime gerek duyulur.

Bir geometrik öğenin rijid bir gövde olduğu düşünülür ve dönüşüm esnasında üzerindeki her nokta aynı dönme ve çeviriye tabi tutulur. Dolayısıyla bunun dönüşümü, üzerindeki bir nokta ile açıklanabilir.

Uzaydaki bir nokta, P (Şekil 7-6) üç koordinatı ile açıklanabilir.

$$(x, y, z)$$

veya matris gösteriminin kullanılması ile, bir sütun matrisi ile:

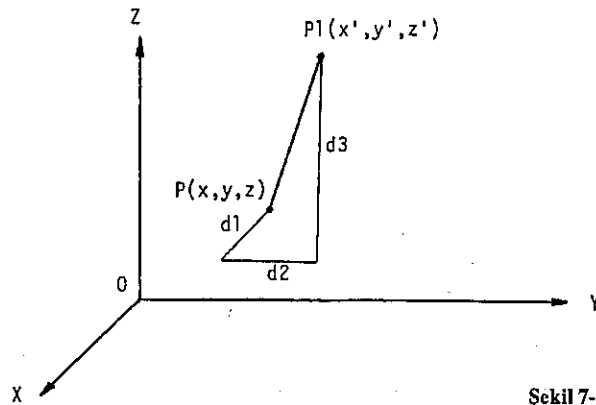
$$[P] = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

P noktasının P1 (x', y', z') noktasına dönüşümü şöyle açıklanabilir:

$$[P1] = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} d1 \\ d2 \\ d3 \end{bmatrix} = [P] + [D] \quad (1)$$

burada $[D]$ çeviri vektörü ve $d1, d2$ ve $d3$ sırası ile X, Y ve Z yönlerinde artış mesafeleridir.

Matematiksel olarak, bir noktanın, $P(x, y, z)$ bir açıda, a koordinat sisteminin orijininin etrafında dönmesi, noktanın konumu değişmeksizin koordinat sisteminin bir açıda, $-a$, dönmesine eşittir. Referans koordinat sistemi $X-Y-Z$ ve X, Y, Z yönlerindeki birim vektörlerde sırası ile i, j ve k olsun.



Şekil 7-6 P noktasının P1 noktasına dönüşümü

Dönmeden sonra, birim vektörleri i', j' ve k' 'nin yön kosinüsleri ile birlikte yeni bir koordinat sistemi, $X'-Y'-Z'$, türetilir ve bu

$$\begin{aligned} C11 &= \cos(X, X') & C12 &= \cos(X, Y') & C13 &= \cos(X, Z') \\ C21 &= \cos(Y, X') & C22 &= \cos(Y, Y') & C23 &= \cos(Y, Z') \\ C31 &= \cos(Z, X') & C32 &= \cos(Z, Y') & C33 &= \cos(Z, Z') \end{aligned}$$

ye eşittir. Burada $(X, X'), (X, Y'), \dots, (Z, Z')$ sırası ile X ekseninden X' eksenine, X ekseninden Y' eksenine, \dots , ve Z ekseninden Z' eksenine saatin tersi yönünde pozitif açılarını temsil ederler. $C11, C21, C31, \dots, C33$ 'ün, X, Y ve Z eksenlerinde birim vektörlerinin (i, j' ve k') izdüşümleri olduğuna da dikkat edilmelidir.

Açısal bir dönüşümü tanımlayan denklem*:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} C11 & C21 & C31 \\ C12 & C22 & C32 \\ C13 & C23 & C33 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [T] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2)$$

* Gellert, W, et.al., The VNR Concise Encyclopedia of Mathematics, pp.533-535, New York: Van Nostrand Reinhold Co., 1975.

burada (x, y, z) ve (x', y', z') sırası ile $X-Y-Z$ ve $X'-Y'-Z'$ koordinat sistemlerindeki aynı noktanın, örneğin P , koordinatlarıdır ve $[T]$ dönüşüm matrisidir. Böylece, verilen orijinal koordinat sistemindeki, $X-Y-Z$, bir noktanın koordinatları ve koordinat sisteminin açısal dönüşümü, aynı noktanın koordinatları yukarıdaki 2 no.lu denklem ile tespit edilebilir. Aynı denklem farklı değerlendirilebilir. Koordinat sistemi değişmeksizin, dönmeden sonraki P noktasının yeni konumunu hesaplamak için bir denklem olarak da düşünülebilir. Yukarıda belirtildiği gibi, bu durumda noktanın döndürülmesi koordinat sisteminin $X-Y-Z$ 'den $X'-Y'-Z'$ 'ne döndürülmesinin tersidir; $[T]$ dönüşüm matrisindeki katsayılar uygun olarak değiştirilmelidir.

Aşağıdaki değişim matrisleri ile Z, X ve Y eksenleri etrafında bir noktanın bir açıda, a , dönmesi verilmektedir (Şekil 7-7):

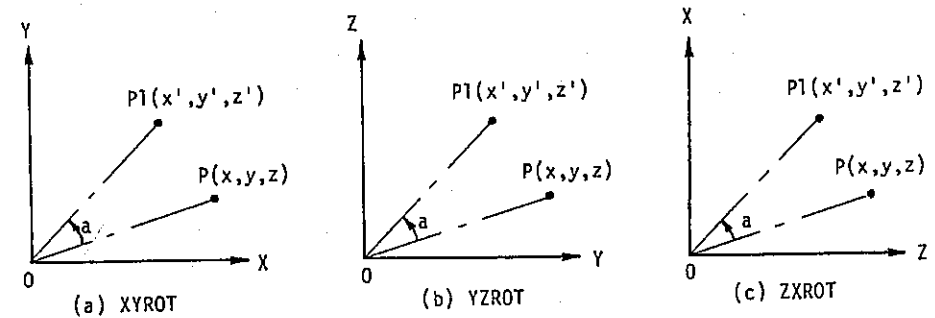
$$\begin{bmatrix} \cos(a) & -\sin(a) & 0 \\ \sin(a) & \cos(a) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \cos(a) & 0 & \sin(a) \\ 0 & 1 & 0 \\ -\sin(a) & 0 & \cos(a) \end{bmatrix} \quad (5)$$

Hem çeviri hem de dönmeyi içine alan genel dönüşüm şu denklemle tanımlanabilir:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} C11 & C21 & C31 \\ C12 & C22 & C32 \\ C13 & C23 & C33 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} d1 \\ d2 \\ d3 \end{bmatrix} \quad (6)$$



Şekil 7-7 Z(a), X(b) ve Y(c) eksenlerinde bir noktanın, P, bir açıda, a, dönmesi.

Bu yukarıda 1 ve 2 no.lu denklemlerin birleşimidir ve matris biçiminde şöyle gösterilebilir:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} C11 & C21 & C31 & d1 \\ C12 & C22 & C32 & d2 \\ C13 & C23 & C33 & d3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (7)$$

veya

$$[P1] = [T_d] [P] \quad (8)$$

burada $[T_d]$ afin veya yeknesak dönüşüm diye adlandırılır. $[T_d]$ dönüşüm matrisindeki alt matrislerin sırası ile dönme ve çeviri dönüşümlerini gösterdiklerine dikkat edilmelidir:

$$\begin{bmatrix} C11 & C21 & C31 \\ C12 & C22 & C32 \\ C13 & C23 & C33 \end{bmatrix} \text{ ve } \begin{bmatrix} d1 \\ d2 \\ d3 \end{bmatrix}$$

7.6.2 APT'de Matris Tanımları

APT'de bir dönüşüm matrisi aşağıdaki deyimle tanımlanır:

MATRIX/.....

SD işlemcisinde bir geometrik öge olarak saklanır. Onun için matris deyimi tekrar tekrar icra edilemez ve aşağıdaki deyim belirlenmeden önce matris sembolü tekrar tanımlanamaz.

CANON/ON

Bir geometrik öğenin değişimi için birkaç değişim matrisi gerektiğinde yer değiştirme kuralı matrislerin sonucuna uygulanmadığı için bu matrislerin uygulama sırası çok önemlidir. MATRIX deyiminin formatları ve kullanımları Tablo 7-3'te listelenmiştir. 1-8 arasındaki MATRIX deyimleri genellikle tek bir koordinat sisteminde bir geometrik öge veya kesici yolunun bir konumdan diğerine dönüşümünü tanımlamak için kullanılır. 9 ve 10 no.lu deyimler, MATRIX deyimi ile belirlenen yerel koordinat sisteminde tanımlanmış bir geometrik öge veya kesici yolunu taban koordinat sisteminde yeniden tanımlamak için kullanılır.

APT-AC SD işlemcisinde, bir APT MATRIX deyimi sadece bir matrisin matematiksel tanımıdır. PATTERN ve aşağıdaki Bölüm 7.7'de açıklanan dönüşüm

deyimleri gibi belli APT deyimlerinde kullanılana kadar geometrik öğelerin ve kesici yolunun dönüşümünü türetmez.

TABLO 7-3 MATRIS DEYİMLERİNİN FORMATLARI VE KULLANIMI

No.	Deyim Formatı	Matematiksel İfade
-----	---------------	--------------------

1 Çeviri formatı

$$\text{MATRIX/TRANSL},d1,d2,[d3] \quad \begin{bmatrix} 1 & 0 & 0 & d1 \\ 0 & 1 & 0 & d2 \\ 0 & 0 & 1 & d3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Yorum $d1$, $d2$ ve $d3$ miktarları bir geometrik öğenin (veya kesici yolunun) ilk konumundan, P, dönüşüm konumuna, P1, X,Y ve Z yönlerindeki artışlı mesafelerdir (Şekil 7-6'ya bakınız) $d3$ mesafesi atlanırsa sıfır alınır.

Örnekler

M1=MATRIX/TRANSL,1,2,3

veya yukarıdaki deyimde $d1$ ve $d2$ 'ye 1 ve 2 atandıysa

M1=MATRIX/TRANSL,d1,d2,3

2 Dönme matrisi

$$\text{MATRIX} \begin{bmatrix} \text{XYROT} \\ \text{YZROT} \\ \text{ZXROT} \end{bmatrix},a \quad \begin{bmatrix} \cos(a) & -\sin(a) & 0 & 0 \\ \sin(a) & \cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{XYROT için})$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) & 0 \\ 0 & \sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{YZROT için})$$

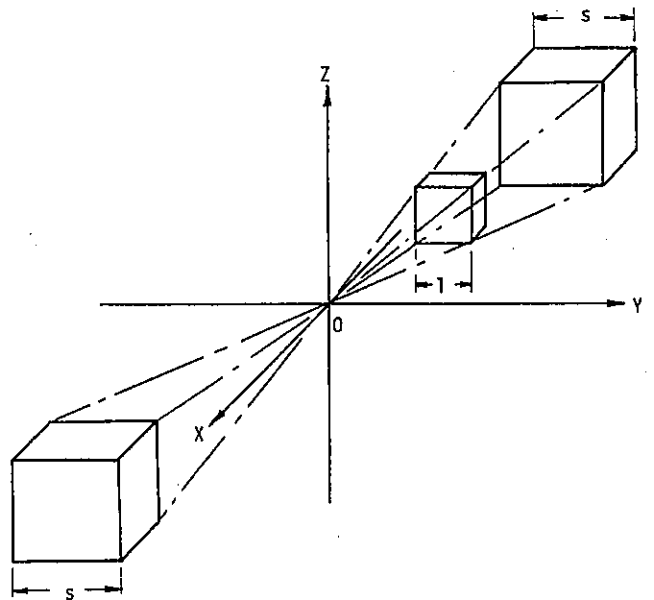
$$\begin{bmatrix} \cos(a) & 0 & \sin(a) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(a) & 0 & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{ZXROT için})$$

Yorum Bu matris, geometrik öğenin, bir açıda, a , Z (XYROT için), X (YZROT için) veya Y (ZXROT için) koordinat eksenleri etrafında dönmesini tanımlar. Açının pozitif yönü sağ el kuralına tabidir.

Örnek

M1=MATRIX/XYROT,45

TABLO 7-3 MATRİS DEYİMLERİNİN FORMATLARI VE KULLANIMI (devamı)

No.	Deyim Formatı	Matematiksel İfade
3	Ölçek matrisi MATRIX/SCALE, s	$\begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
		
	<p>Yorum Bu matris bir geometrik öğeyi veya kesici yolunu s çarpanı ile büyültür veya küçültür. Yukarıdaki şekilden görülebileceği gibi büyültme (s, 1'den büyük) veya küçültme (s, 1'den küçük) koordinat sisteminin orijinine göredir. $s = -1$ olduğunda, dönüşüm orijin aracılığı ile bir yansımadır. s sıfırdan küçük olduğunda</p> $s = (-1) \cdot s $ <p>olur. Böylece, sıfırdan küçük herhangi bir s değeri için ölçek matrisi tarafından tanımlanan dönüşüm matrisi koordinat sisteminin orijini aracılığı ile (bir s çarpanı ile) büyütülmüş bir yansımadır.</p>	
4	Bir veya daha fazla koordinat düzlemleri etrafında ayna görüntüsü dönüşümü MATRIX/MIRROR, {XYPLAN, YZPLAN, ZXPLAN} \$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (MIRROR, XYPLAN)

TABLO 7-3 MATRİS DEYİMLERİNİN FORMATLARI VE KULLANIMI (devamı)

No.	Deyim Formatı	Matematiksel İfade
	$\left[\begin{array}{l} \text{XYPLAN} \\ \text{YZPLAN} \\ \text{ZXPLAN} \end{array} \right] \left[\begin{array}{l} \text{XYPLAN} \\ \text{YZPLAN} \\ \text{ZXPLAN} \end{array} \right]$	$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (MIRROR, YZPLAN)
		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (MIRROR, ZXPLAN)
		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (MIRRORXYPLANZXPLAN veya MIRRORZXPLANXYPLAN)
		$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (MIRRORXYPLANYZPLAN veya MIRRORYZPLANXYPLAN)
		$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (MIRRORYZPLANZXPLAN veya MIRRORZXPLANYZPLAN)
		$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (MIRRORXYPLANYZPLAN, ZXPLAN)
5	Genel dönüşüm matrisi MATRIX/c11,c21,c31,d1,c12,c22,\$ c32,d2,c13,c23,c33,d3	$\begin{bmatrix} c11 & c21 & c31 & d1 \\ c12 & c22 & c32 & d2 \\ c13 & c23 & c33 & d3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
6	Bir doğru (L1) veya düzlem (PL1) etrafında yansıyan görüntü dönüşümü MATRIX / MIRROR, {L1, PL1}	

Yorum Bir koordinat düzlemine göre ayna dönüşümü bu düzleme dik olan eksen yönündeki koordinatın işaretini değiştirmekten ibarettir. Örneğin, bir nokta (a, b, c) , $Y-Z$, $Z-X$ ve $X-Y$ düzlemlerine göre yansıyan dönüşümleri ile $(-a, b, c)$, $(a, -b, c)$ ve $(a, b, -c)$, konumuna alınabilir. Bu deyimde birden fazla yansıma işlemi birleştirilebilir ve onların bu deyimdeki sırası sonucu etkilemez.

TABLO 7-3 MATRİS DEYİMLERİNİN FORMATLARI VE KULLANIMI (devamı)

No.	Deyim Formatı	Matematiksel İfade

Yorum APT'de bir doğru, onun üzerinden geçen ve X-Y düzlemine dik bir düzlemdir. Onun için esasen bu deyim bu düzlem etrafında bir yansıyan-görüntü dönüşümünü tanımlar.

7 Birleşik dönüşüm matrisi

$$\text{MATRIX} \left\{ \begin{array}{l} M1 \\ \text{XYROT}_{a1} \\ \text{YZROT}_{a2} \\ \text{ZXROT}_{a3} \\ \text{SCALE}_{s1} \\ \text{TRANSL}_{d1, d2, d3} \end{array} \right\} \cdot \$ [M2][M1]$$

$$\text{TIMES} \left\{ \begin{array}{l} M2 \\ \text{XYROT}_{a4} \\ \text{YZROT}_{a5} \\ \text{ZXROT}_{a6} \\ \text{SCALE}_{s2} \\ \text{TRANSL}_{d4, d5, d6} \end{array} \right\}$$

Yorum M1 ve M2 iki tanımlanan matris için sembollerdir. TIMES kelimesinden sonraki matris işlemini her zaman önce uygulanır.

Örnekler

(1) Şu matrisle tanımlanan dönüşüm

$$M1 = \text{MATRIX/XYROT}_{,a1, \text{TIMES,TRANSL}_{,d1, d2, d3}}$$

bir çeviri ($d1, d2, d3$)dir; bunu Z eksenine etrafında $a1$ açısıyla dönme takip etmektedir.

(2) Aşağıdaki matrisle tanımlanan dönüşüm

$$M2 = \text{MATRIX/TRANSL}_{,d1, d2, d3, \text{TIMES,XYROT}_{,a1}}$$

dönmeyi takip eden bir çeviridir. Örnek 1'deki M1 matrisi ile tanımlanandan farklıdır.

TABLO 7-3 MATRİS DEYİMLERİNİN FORMATLARI VE KULLANIMI (devamı)

No.	Deyim Formatı	Matematiksel İfade
8	Ters dönüşüm matrisi	
	MATRIX/INVERS,M1	$M1^{-1} = \begin{bmatrix} c11 & c21 & c31 & d1 \\ c12 & c22 & c32 & d2 \\ c13 & c23 & c33 & d3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
		M1'in ters matrisi
		$M1^{-1} = \begin{bmatrix} c11 & c12 & c13 & -(c11 \cdot d1 + c12 \cdot d2 + c13 \cdot d3) \\ c21 & c22 & c23 & -(c21 \cdot d1 + c22 \cdot d2 + c23 \cdot d3) \\ c31 & c32 & c33 & -(c31 \cdot d1 + c32 \cdot d2 + c33 \cdot d3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Yorum Eğer M matrisi bir noktayı A'dan B'ye dönüştürürse, yukarıdaki deyimle tanımlanan ters dönüşüm B konumundaki deyimle tanımlanan ters dönüşüm B konumundaki bir noktayı A konumuna dönüştürür. Birkaç matrisin birleşimi olan M1 matrisinin tersi aşağıdaki denklemlerle açıklanabilir.

$$\text{Eğer } (M1) = (M1') \cdot (M2') \text{ ise } (M1)^{-1} = (M2')^{-1} \cdot (M1')^{-1}$$

$$\text{Eğer } M1 = (M1') \cdot (M2') \cdot (M3') \text{ ise } (M1)^{-1} = (M3')^{-1} \cdot (M2')^{-1} \cdot (M1')^{-1}$$

$$\text{Eğer } M1 = (M1') \cdot (M2') \cdot \dots \cdot (Mn') \text{ ise } (M1)^{-1} = (Mn')^{-1} \cdot \dots \cdot (M2')^{-1} \cdot (M1')^{-1}$$

9 PL1, PL2 ve PL3 üç düzlemi ile tanımlanan matris

MATRIX/PL1,PL2,PL3

Eğer PL1, PL2 ve PL3 şu denklemlerle tanımlanırsa

$$(a1)x + (b1)y + (c1)z - d1 = 0$$

$$(a2)x + (b2)y + (c2)z - d2 = 0$$

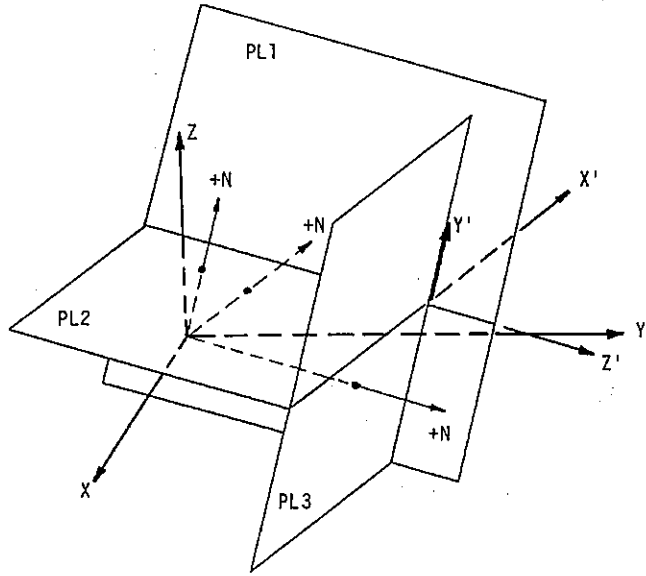
$$(a3)x + (b3)y + (c3)z - d3 = 0$$

tanımlanan matris bu olur

$$\begin{bmatrix} a1 & a2 & a3 & a1 \cdot d1 + a2 \cdot d2 + a3 \cdot d3 \\ b1 & b2 & b3 & b1 \cdot d1 + b2 \cdot d2 + b3 \cdot d3 \\ c1 & c2 & c3 & c1 \cdot d1 + c2 \cdot d2 + c3 \cdot d3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

N: düzlemlere dik

TABLO 7-3 MATRİS DEYİMLERİNİN FORMATLARI VE KULLANIMI (devamı)

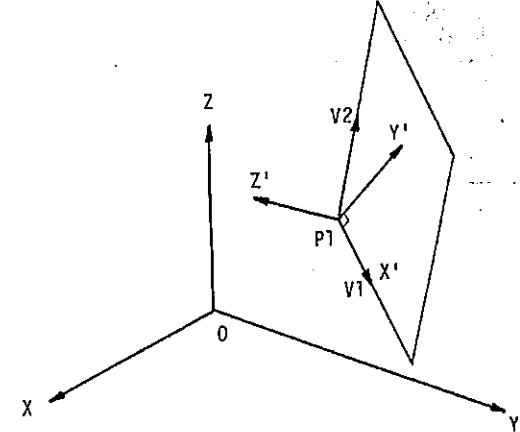
No.	Deyim Formatı	Matematiksel İfade
		

Yorum X-Y-Z taban koordinat sisteminde tanımlanan üç PL1, PL2 ve PL3 düzlemleri X'-Y'-Z' yerel koordinat sisteminin sırası ile Y'-Z', Z'-X' ve X'-Y' koordinat düzlemleridir. Bir geometrik öğe veya kesici yolunun X'-Y'-Z' koordinat sisteminde tanımlanacağı farz edilir. Tanımlanan matris geometrik öğeyi X-Y-Z koordinat sisteminde tanımlanan öğeye dönüştürür. Geometrik öğenin uzaydaki konumu değişmezken bu durumda geometrik öğeyi tanımlamak için koordinat sisteminin yerelden taban sistemine değiştirildiğine dikkat ediniz. Dönüşümden sonra geometrik öğe, taban koordinat sisteminde açıklanmaktadır. X', Y' ve Z' eksenlerinin verilen üç düzleme pozitif yönde dik olduğuna da dikkat ediniz. Böylece bu üç düzlemin tanımı bunların normallerinin sağ-el koordinat sistemi oluşturacak şekilde olmalıdır.

10 Bir nokta, P1, ve iki vektör, V1 ve V2 ile tanımlanan matris

MATRIX/P1,V1,V2

TABLO 7-3 MATRİS DEYİMLERİNİN FORMATLARI VE KULLANIMI (devamı)

No.	Deyim Formatı	Matematiksel İfade
		

Yorum Bu matris, orijinin P1 noktasında ve X' ekseninin V1 vektörü ile tanımlandığı bir yerel koordinat sistemini, X'-Y'-Z', tanımlar. Yerel sistemin Y' eksenine ve V1 ve V2 vektörleri ile tanımlanan düzleme diktir. V2 ve Y' arasında kalan açı 90 dereceden daha az olmalıdır (Yerelden X-Y-Z taban sistemine olan). Koordinat sistemi değişimi bu deyim özelliği aracılığı ile gerçekleştirilebilir. Yukarıdaki 9 nolu deyim ile aynı kullanıma sahiptir.

7.7 DÖNÜŞÜM MATRİSİ İLE GEOMETRİK ÖGELERİN VE KESİCİ YOLLARININ TANIMLANMASI

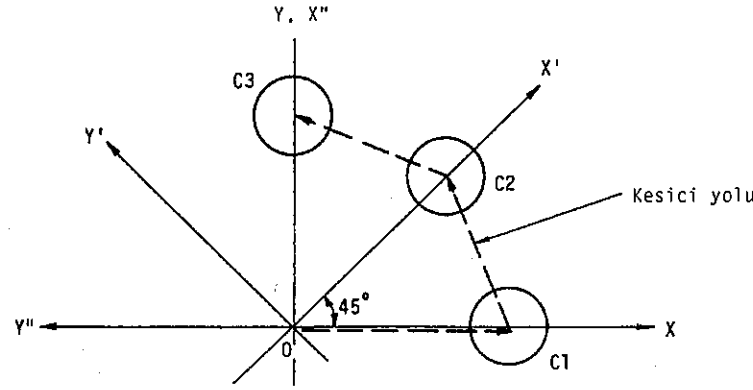
Bir APT programında tanımlanan geometrik öğeleri ve kesici yolunu idare etmek için üç deyim kullanılır: REFSYS, TRACUT ve COPY. İlk deyim, REFSYS, ya bir geometrik öğeyi tanımlamak için en uygun koordinat sistemini kullanmamıza veya taban koordinat sisteminde tanımlanan bir geometrik öğeyi dönüştürmemize izin verir. İkinci deyim, TRACUT, CLDATA'nın veya kesici yolunun değişimi için kullanılır. Üçüncüsü bir kesici yolunu kopyalamamıza ve değişimine izin verir. Bu deyimlerin detaylı açıklaması ve kullanımları aşağıda verilmektedir.

7.7.1 REFSYS Deyimi ile Geometrik Öğelerin Dönüşümü

Bazen bir geometrik öğeyi taban koordinat sisteminde (işleme operasyonunda kullanılan koordinat sistemi) tanımlamak zor, fakat seçilen bir koordinat sisteminde

tanımlamak kolay olabilir. Önce bu geometrik öğeyi seçilen koordinat sisteminde tanımlayabilirsiniz. Sonra taban koordinat sisteminde REFSYS deyimini kullanarak onu tekrar tanımlayabiliriz. REFSYS deyimini, taban koordinat sisteminde tanımlanan geometrik öğelerin değişiminde de kullanılabilir.

REFSYS deyiminin formatı, anlamı ve kullanımları Tablo 7-4'te gösterilmektedir. Görüleceği gibi, REFSYS/(matris sembolü) ve REFSYS/NOMORE deyimleri bir programdaki dönüşümün etkili aralığını tanımlar. Dönüşüme tabi tutulacak bu geometrik öğeler bu iki deyim arasında tanımlanmalıdır. Bir APT programı işleme tabi tutulurken ve REFSYS/ deyimleri okunurken matris ve geometrik tanımlama deyimleri NC işlemci tarafından hafızada saklanır ve bu aşamada dönüşüm gerçekleştirilmez. Dönüşüm, sadece bu geometrik öğeler REFSYS/NOMORE deyiminden sonraki deyimlere atıfta bulunulduğunda gerçekleştirilir. Bu



APT Program

```

REFSYS/(M1=MATRIX/XYROT, 45)
C2=CIRCLE/1, 0, 0.2
A1) REFSYS/NOMORE
REFSYS/(M2=MATRIX/XYROT, 90)
C3=CIRCLE/1, 0, 0.2
REFSYS/NOMORE
C1=CIRCLE/1, 0, 0.2
FROM/0, 0, 0
GOTO/(POINT/CENTER, C1)
GOTO/(POINT/CENTER, C2)
GOTO/(POINT/CENTER, C3)

```

Şekil 7-8 REFSYS deyiminin uygulaması

TABLO 7-4 REFSYS DEYİMİNİN FORMATI, ANLAMI VE KULLANIMI

Format	Anlamı ve Kullanımı
REFSYS/M1\$\$ M1 IS A MATRIX	İki farklı kullanımı vardır:
REFSYS/NOMORE	(1) M1 matrisi X-Y-Z taban koordinat sisteminin X'-Y'-Z' yerel koordinat sistemine değişimini tanımlar. REFSYS/M1 ve REFSYS/NOMORE deyimleri arasında tanımlanan geometrik öğeler yerel koordinat sisteminde tanımlanır.
	Her ne zaman REFSYS/NOMORE'den sonra bir deyimde bu geometrik öğelerden herhangi birine atıfta bulunulursa o, taban koordinat sisteminde tekrar tanımlanır.
	(2) Bu matris, taban koordinat sisteminde geometrik öğelerin değişimini tanımlar.
	İki REFSYS deyimini arasında tanımlanan geometrik öğeler taban koordinat sisteminde tanımlanır.
	Her ne zaman bu geometrik öğelerden herhangi birisine REFSYS/NOMORE'dan sonra bir deyimde atıfta bulunulursa o, taban koordinat sisteminde matrisle dönüşüme tabi tutulur.
	Not: REFSYS/M1 deyiminden önce ve REFSYS/NOMORE deyiminden sonra tanımlanan geometrik öğeler dönüşüme tabi tutulmazlar.

dönüşümün sonucu, yerel koordinat sisteminde tanımlanan geometrik öğeleri taban koordinat sisteminde tekrar tanımlamaktır. Geometrik öğeler taban koordinat sisteminde tanımlanmışsa, bunlar sadece REFSYS deyiminde belirlenen matrisle göre dönüşüme tabi tutulurlar.

Örneğin, Şekil 7-8'de gösterilen APT Programı aynı parametre gruplu üç C1, C2 ve C3 çemberini tanımlar. C2 ve C3 sırası ile X'-Y' ve X''-Y'' dönüşüm koordinat sistemlerinde tanımlanmış geometrik öğeler olarak düşünülebilir. C2 ve C3 çemberleri REFSYS/NOMORE'dan sonraki deyimlerde atıfta bulunulduğunda X-Y taban koordinat sisteminde M1 ve M2 tanımlanan dönüşümlere ters olan koordinat

sistemlerin dönüşümleri aracılığı ile bunlar tekrar tanımlanır. Böylece, REFSYS deyimini geometrik öğeleri tanımlamak için seçilmiş uygun bir koordinat sistemini kullanmamıza izin verir. Tam tersine, biz C2 ve C3 çemberlerini, taban koordinat sisteminde tanımlanan geometrik öğeler olarak düşünebiliriz. Kendilerine atıfta bulunulduğunda bunlar, taban koordinat sisteminde sırası ile 45 ve 90 derece döndürülürler.

Şekil 7-8'de gösterilen APT programında A1 etiketli deyim kullanılmayabilir. Çünkü bir sonraki deyim, REFSYS/M2 önceki değişimin sonucunu REFSYS/M1 ve yeni dönüşümün REFSYS/M2 başlangıcını belirler. M1 ve M2, X-Y-Z'ye göre tanımlanmalıdır. Şekil 7-9 Şekil 7-8'de gösterilen programın bilgisayar çıktısını temsil eder. Okuyucu dönüşümü doğrulamak için CLDATA'yı kontrol etmelidir.

REFSYS deyimini tarafından etkilenen geometrik öğeler şunlardır: Nokta, vektör, PNTVCT, SPLAIN, desen, doğru, düzlem, çember, elips, hiperbol, GCONIC, LCONIC, küre, koni, silindir, torus, RLDSRF ve QUADRIC deyimleriyle tanımlanmış yüzeyler.

7.7.2 TRACUT Deyimi ile bir Kesici Yolunun Dönüşümü

REFSYS deyiminde olduğu gibi, TRACUT deyimini de ikili kullanılmalıdır: TRACUT/(matris sembolü) ve TRACUT/NOMORE. Bu deyim kullanımı aşağıdaki gibidir:

TRACUT/(matris sembolü)

.....
.....
.....
.....
.....
.....

hareket deyimleri

TRACUT/NOMORE

```
ISN 00001 PART NO TEST REFSYS
ISN 00002 CLPRNT
ISN 00003 FEDRAT / 2.0
ISN 00004 CUTTER / 0.5
ISN 00005 C1=CIRCLE / 1, 0, 0, 0.2
ISN 00006 REFSYS/(M1=MATRIX/XYROT, 45)
ISN 00007 C2=CIRCLE/1, 0, 0.2
ISN 00008 REFSYS/(M2=MATRIX/XYROT, 90)
ISN 00009 C3=CIRCLE/1, 0, 0.2
```

```
ISN 00010 REFSYS/NOMORE
ISN 00011 FROM/0, 0, 0
ISN 00012 GOTO/(POINT/CENTER, C1)
ISN 00013 GOTO/(POINT/CENTER, C2)
ISN 00014 GOTO/(POINT/CENTER, C3)
ISN 00015 END
ISN 00016 FINI
NO DIAGNOSTICS ELICITED DURING TRANSLATION PHASE
16 N/C SOURCE RECORDS (SYSIN)
.... SECTION 3 ....
ISN
0001 PARTNO/ TEST REFSYS
0003 FEDRAT/ 2.00000000
0004 CUTTER/ 0.50000000
0011 FROM/
0.0 0.0 0.0
0012 GOTO/
1.00000000 0.0 0.0
0013 GOTO/
0.70710678 0.70710678 0.0
0014 GOTO/
0.0 1.00000000 0.0
0015 END
0016 ***** FINI *****
```

Şekil 7-9 Şekil 7-8'de gösterilen programın bilgisayar çıktısı.

İki TRACUT deyimini arasında belirlenen kesici yolu taban koordinat sisteminde tanımlanır ve TRACUT deyiminde belirlenen matrise göre aynı koordinat sisteminde değişime tabi tutulur. Örneğin (Şekil 7-10).

TRACUT/(M2=MATRIX/XYROT,90)

deyimini Şekil 7-9'da verilen APT programına eklersek, kesici yolu 90 derece ile saatin tersi yönünde döndürülecektir (Şekil 7-10'daki CLDATA'ya bakınız). TRACUT/NOMORE deyimini önceden tanımlanan bir TRACUT deyimini iptal etmek için kullanılır.

Eğer TRACUT/NOMORE deyiminin müdahalesi olmaksızın bir TRACUT/M1 deyimini (M1 bir matrisin sembolüdür) bir TRACUT/M2 deyimini takip ederse iki TRACUT deyimleri arasında gözükken kesici yolu, veya CLDATA, M1 ile ve TRACUT/M2 deyiminden sonraki CLDATA, M2 ile değişime tabi tutulur. Bu, ikinci TRACUT deyiminin birincisini otomatikman iptal ettiği anlamına gelir.

TRACUT deyiminde tanımlanan bir değişim, bir diğerine bölü işareti ve matris

```

ISN 00001 PARTNO TEST REFSYS
ISN 00002 CLPRNT
ISN 00003 FEDRAT / 2.0
ISN 00004 CUTTER / 0.5
ISN 00005 C1=CIRCLE / 1, 0, 0, 0.2
ISN 00006 REFSYS/(M1=MATRIX/XYROT, 45)
ISN 00007 C2=CIRCLE/1, 0, 0, 0.2
ISN 00008 REFSYS/(M2=MATRIX/XYROT, 90)
ISN 00009 C3=CIRCLE/1, 0, 0, 0.2
ISN 00010 REFSYS/NOMORE
ISN 00011 FROM/0, 0, 0; TRACUT/(MATRIX,XYROT, 90)
ISN 00013 GOTO/(POINT/CENTER, C1)
ISN 00014 GOTO/(POINT/CENTER, C2)
ISN 00015 GOTO/(POINT/CENTER, C3); TRACUT/NOMORE
ISN 00017 END
ISN 00018 FINI
NO DIAGNOSTICS ELICITED DURING TRANSLATION PHASE
16 N/C SOURCE RECORDS (SYSIN)
... SECTION 3 ...
ISN
0001 PARTNO/ TEST REFSYS
0003 FEDRAT/          2.00000000
0004 CUTTER/          0.50000000
0011 FROM/
0012 TRACUT/
0013 GOTO/
0014 GOTO/
0015 GOTO/
0016 TRACUT/
0017 END
0018 ***** FINI *****
... END OF SECTION 3 ...

```

Şekil 7-10 TRACUT deyiminin kullanımını gösteren bir APT programı. İki TRACUT deyiminin eklendiği bu program Şekil 79'daki ile aynıdır.

sembolü arasına LAST düzelticisi konarak eklenebilir:

TRACUT/LAST,M1

Aşağıdaki APT programında bir örnek verilmektedir:

```

.....
TRACUT/M0
.....
..... } (CLDATA M0 ile değişime tabi tutuluyor)
.....
TRACUT/LAST,M1
.....
..... } (CLDATA önce M0 ve sonra M1 ile değişime
tabi tutuluyor)
TRACUT/M2
.....
..... } (CLDATA önce M2 ve sonra M1 ile değişime
tabi tutuluyor)
.....
TRACUT/LAST,NOMORE
.....
..... } (CLDATA sadece M2 ile değişime tabi tutuluyor)
TRACUT/NOMORE
.....
..... } (değişim yok)
.....

```

TRACUT/LAST deyiminde belirlenmiş dönüşümün ya TRACUT/LAST deyiminden önce veya sonra belirlenmiş TRACUT deyimini ile tanımlanan dönüşüme eklendiğine dikkat ediniz. Bir LAST düzelticili TRACUT deyimini, sadece, onu takip eden LAST düzelticili bir TRACUT deyimini ile iptal edilebilir (ya TRACUT/LAST, [matris sembolü] veya TRACUT/LAST, NOMORE). Ayrıca, LAST düzelticili bir TRACUT deyimini LAST düzeltici olmaksızın bir önceki TRACUT deyimini iptal edemez.

TRACUT dönüşüm deyimini ile tanımlanan kesici yolunun iki kademede gerçekleştirildiğinin belirtilmesi gerekir. TRACUT ve TRACUT/NOMORE deyimleri arasında tanımlanan kesici yolu dönüşüm olmaksızın, önce NC işlemcisi hesaplama kısmında* hesaplanır. Sonra, NC işlemcisi düzeltme (edit) kısmında

* IBM APT-AC SD işlemcisinin yapısı için Bölüm 8.5'e bakınız.

dönüşüme tabi tutulur ve TRACUT deyiminde belirlenmiş matrise göre CLDATA olarak çıktısı verilir. Fakat, NC işlemcisi, bir sonraki kesici hareketinin hesaplanması için referans veya türetilen son kesici konumunu kullanır. Böylece hemen dönüşümden sonraki kesici hareketini tespit etmek için, olan referans konumu son dönüşüme uğramamış kesici konumudur. Bu aşağıdaki örnekle açıklanabilir.

```
PARTNO
L1=LINE/XAXIS,1
L2=LINE/YAXIS,-1
M1=MATRIX/XYROT,45
CUTTER/1.0
FEDRAT/1.0
FROM/1,0,0
TRACUT/M1
      GO/ON,L1
TRACUT/NOMORE
GO/ON,L2
END
FINI
```

Bu program GO/ON, L1 deyiminin icrasından sonra kesicinin (1,0,0) noktasından (0,1.4142,0) noktasına hareketini tanımlanır. Fakat, SD işlemcisinde kesici konumu, yani (1,1,0), son kesici konumu olarak gözönünde tutulur ve bir sonraki hareketi, tespit etmek için referans noktası olarak kullanılır. Sonuç olarak, GO/ON,L2 deyimi ile tanımlanan bitiş konumu (-1,1.4142,0) yerine (1,1,0)'dır.

7.7.3 COPY Deyimi İle Bir Kesici Yolunu Tanımlama

İsminden de anlaşılacağı üzere, COPY deyimi önceden tanımlanan kesici yolunu kopyalama veya hem kopyalama hem de dönüşümü için kullanılabilir. Kopyalanan kesici yolu INDEX/i ve INDEX/i, NOMORE deyimleri arasında belirlenmelidir. Aynı i indis numaralı, INDEX/i NOMORE deyiminden sonra gelen bir COPY deyimi kesici yolunun o parçasının dönüşüm ve kopyalanması için APT-AC SD işlemcisini yönlendirir.

Bir COPY deyiminde dönüşümü tanımlamak için seçilebilen bir kaç düzeltici vardır. COPY deyiminin söz dizimi:

```
COPY//, {
  SAME
  TRANSL,x,y,z
  XYROT,a
  YZROT,a
  ZXROT,a
  MODIFY,M1
  SCALE,s
  P1, {
    ZXROT
    XYROT
    YZROT
  }, a[, ROTREF]
}
```

burada n , türetilcek olan kesici yolunun ek kopyalarının sayısını tanımlar. İlk kopya, tanımlanan kesici yoluna dayalı olarak ve n 'nci kopyanın ($n-1$)nci kopyasına bağlı olarak yapıldığına dikkat edilmelidir. Bu düzeltici ve parametrelerin kullanımları ve anlamları aşağıdaki gibidir:

SAME

Bu düzeltici, sadece kesici yolunun kopyalanacağını ve dönüşümün yer almadığını belirler.

Örnek

```
.....
.....
INDEX/2
GOTO/0,1,0
GOTO/0,2,0
.....
GOTO/3,4,5
INDEX/2,NOMORE
..... } (A)
.....
COPY/2,SAME,1
.....
.....
```

INDEX deyimleri arasındaki kesici yolu bir kere kopyalanır ve A kısmındaki deyimlerle tanımlanan kesici yolundan sonra yerleştirilir. Bir kesici yolu, iki INDEX deyimi arasında önceden tanımlananla aynı olduğu zaman bu seçenek faydalıdır.

TRANSL,x,y,z

Bu düzeltici, kopyalandığı zaman kesici yolu verisi X , Y ve Z yönlerinde x , y ve z ile çevirileceğini belirler.

Örnek (Şekil 17-11)

```

.....
INDEX/1
.....
..... } (A)
.....
INDEX/1,NOMORE
COPY/1,TRANSL,3,0,0.2
.....
.....

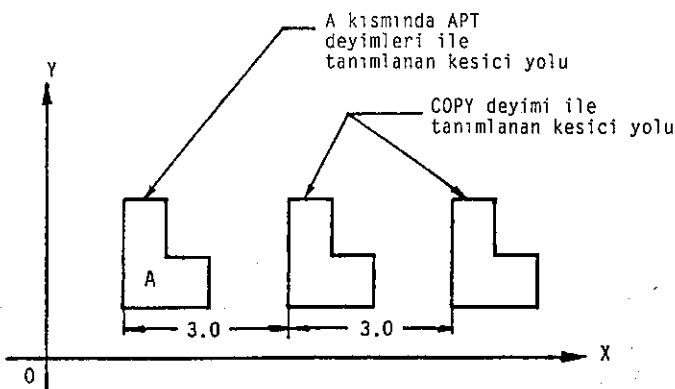
```

Şekil 7-11'de görüleceği gibi, her dönüşüm hemen bir önceki kesici yoluna bağlı olarak gerçekleştirilir; yani ilk kopyalama orjinal kesici yoluna ve belirlenmiş dönüşüme bağlı olarak yapılır. Sonra ikinci kopyalama, ilk kopyalamanın kesici yolu ve belirlenmiş dönüşüme bağlı olarak oluşturulur.

XYROT, a ; YZROT, a ; ZXROT, a

Bu düzelticiler, sırası ile Z , X ve Y eksenleri etrafında bir a açısında dönmeyi tanımlar. Eğer birden çok kopya gerekirse, dönüşüm yukarıda açıklanan kuralları takip eder.

MODIFY,M1; SCALE, s



Şekil 7-11 COPY deyimi ile türetilen kesici yolu

Bu düzelticiler, s ölçek çarpanına (SCALE, s) dayalı olarak koordinat sisteminin orijinine göre büyültmeyi veya küçültmeyi ve M1 matrisine bağlı olarak (MODIFY,M1) dönüşümü belirler.

P1,XYROT, a [,ROTFREF], n ; P1,YZROT, a [,ROTFREF], n ; P1,ZXROT, a [,ROTFREF], n .

Bu düzelticiler ile tanımlanan dönüşümler aşağıdaki gibidir. Dönüşüm yapılmadan önce, (0,0,0) noktasının (yani koordinat sisteminin orijini) dönüşümü yapılacak kesici yolu ile irtibatlandırıldığı kabul edilir. Dönüşüm iki kademedен oluşur: Önce irtibatlanan bu nokta önce belirlenmiş noktaya, P1, hareket ettirilir ve sonra kesici yolu, P1 noktasından geçen ve Z (XYROT için), X (YZROT için) veya Y eksenine (ZXROT için) paralel olan bir eksen etrafında bir a açısında döndürülür. Eğer n , 1'den büyükse, ilk kopyalama, orjinal kesici yolunun P1 noktasına çevrilmesi ve sonra a açısı kadar döndürülmesiyle yapılır. İkinci kopyalama, orjinal kesici yolunun aynı P1 noktasına çevrilmesi ve sonra $2a$ açısında, döndürülmesi, vb. ile yapılır. Eğer bir kopyalama, çevirimsiz kesici yolunun belirlenmiş bir nokta, P1 etrafında döndürülmesi ile yapılırsa, isteğe bağlı ROTREF kelimesi belirlenmelidir.

Kopya deyiminin, INDEX/ i ve INDEX/ i , NOMORE çifte deyimini ile birlikte kullanılması gerektiğini önceden belirtmiştik. Gerçekte, bu her zaman gerekli değildir, bir COPY deyimini, INDEX/ i , NOMORE deyimini olmaksızın INDEX/ i deyiminden sonra belirlenebilir. Bu durumda COPY deyimini, COPY ve INDEX/ i , NOMORE deyimlerinin ikisinin de fonksiyonlarına sahiptir. Şekil 7-12'de gösterilen örnekte belirtildiği gibi COPY deyimleri de iç içe yazılabilir.

7.7.4 REFSYS, COPY ve TRACUT Deyimleri ile ilgili Bazı Yorumlar

REFSYS, COPY ve TRACUT, geometrik öğeler veya kesici hareketi için değişimi tanımlamak gayesiyle kullanılan üç deyimdir. REFSYS deyimini geometrik öğelerin dönüşümü için diğer ikisi kesici yolu dönüşümü için kullanılır. Önceden tanımlanana benzer şekilde bir kesici yolu, daha kolayca ya önce geometrik öğelerin dönüşümü ve sonra, dönüşümü yapılan yüzeyler boyunca kesici yönlendirilmesi ile veya dönüşümü yapılmayan geometrik öğelere dayalı olarak tanımlanan kesici yolunun doğrudan dönüşümü ile tanımlanabilir.

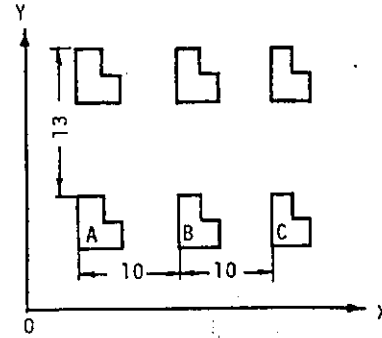
Önceden de belirtildiği gibi, farklı dönüşüm deyimleri kullandığında CLDATA farklı yollarla türetilir. Bu üç dönüşüm deyimlerine karşılık gelen SD işlemcisinde işlemlerin hesaplanması Tablo 7-5'te verilmektedir.

REFSYS deyimini ile tanımlanan dönüşümde, atıfta bulunulduğunda geometrik öğeler dönüşüme tabi tutulur, diğer yandan TRACUT veya COPY deyimleri ile tanımlanan dönüşümler de değişmezler. Ayrıca, NC işlemcisinin hesaplama kıs-

INDEX/1

COPY/1,TRANSL,10,0,0,2

COPY/1,TRANSL,0,13,0,1

A parçası için kesici
yolunu tanımlayan
deyimlerbu kısımda kesici
hareketi yok

Şekil 7-12 İki kopya yapma işlemi için kesici yollarının tanımlanması. İlk COPY deyimini, A konumunda kesici yolunun iki kopyasını yapar, bundan dolayı B ve C kesici yolu elde edilir. İlk COPY çevrimi ikincisinde yerleştirildiği için, A, B ve C, üç kesici yolu dönüşüme tabi tutulur ve kopyalanır.

mında hesaplanan ve saklanan kesici konumları COPY veya TRACUT dönüşümleri tarafından değiştirilmez. Hesaplama kısmında belirlenen son kesici konumu bir sonraki hareketi tespit etmek için referans noktası olarak kullanıldığı için, düzeltme kısmı tarafından verilen son kesici konumu (yani, CLDATA'daki kesici konumu) bir sonraki kesici hareketini hesaplamak için kullanılan referans noktası değildir. TRACUT ve COPY deyimlerinin tekrar tekrar kullanılması veya kesici hareketinin bu deyimlerin herhangi birisinin kullanımı sonucu yanlış tanımlanmasıyla sonuçlanabilir. TRACUT ve COPY deyimleri bir kesici hareketini tanımlamak için kullanıldıklarında, bunların değişimleri Tablo 7-5'te gösterilen sırada uygulanırlar. Aşağıda verilen örnekler, kesici yolunun tanımlanmasında hesaplama işleminin önemini açıklamaktadır.

Şekil 7-14, INDEX çifti deyimleri arasında belirlenen TRACUT deyimindeki LAST düzelticisinin kullanım örneklerini vermektedir. COPY deyiminin amacı, 0-A-B-C kesici yolunu kopyalamak ve 0₁-A₁-B₁-C₁ kesici yolunu elde etmektir (Şekil 7-14). Fakat, COPY deyiminin icrası ile elde edilen kesici yolu, 0₁-A₁-B₂-C₁'dir. Bunun nedeni, LAST düzelticili TRACUT kesici yolu dönüşüm deyiminin kopyalama bitirilene kadar icra edilmeyecek olmasıdır. Onun için, B' (yani CC2 çemberinin merkezi) olarak tanımlanan kesici konumu önce B'' noktasına 90 derece döndürülür, sonra B3 konumuna kopyalanır, dönüştürülür ve son olarak B2 konumuna 180 derece döndürülür.

İkinci örnek, profilli çemberlerle yaklaşık olarak verilen 12 dişli bir dişli çarktır (Şekil 7-15). Listesi verilen A programı, parmak frezeyi profilin sağından gitmek için yönlendirir.

TABLO 7-5 ÜÇ DÖNÜŞÜM DEYİMİ İÇİN SD İŞLEMCİSİNDEKİ HESAPLAMA İŞLEMLERİ

Dönüşüm Deyimi	Hesaplama İşlemi						
REFSYS	<ol style="list-style-type: none"> 1. Dönüşümü yapılacak geometrik öğeleri ve dönüşüm matrisini saklar (APT-AC SD işlemcisinin dönüşüm kısmı tarafından yapılır) 2. Kendilerine atıfta bulunulduğunda, geometrik öğelerin dönüşümünü yapar (hesaplama kısmı tarafından) 3. Dönüşümü yapılan geometrik öğelere bağlı olarak kesici konumunu hesaplar (hesaplama kısmı tarafından) 4. Hesaplama sonuçlarına göre CLDATA'yı türetir. 						
TRACUT	<ol style="list-style-type: none"> 1. Geometrik öğeleri ve TRACUT deyimini ile tanımlanan dönüşüm matrisini saklar (çeviri kısmı tarafından) 2. Dönüşüm olmaksızın TRACUT deyimleri arasında tanımlanan kesici yolunu hesaplar (hesaplama kısmı tarafından) 3. TRACUT deyimlerinde tanımlanan matrislere göre hesaplanan kesici konumunun değişimini yapar ve onun CLDATA olarak çıktısını verir (düzeltme kısmı tarafından) 						
COPY	<ol style="list-style-type: none"> 1. Geometrik öğeleri ve COPY deyimini ile tanımlanan dönüşüm matrisini saklar (çeviri kısmı tarafından) 2. INDEX deyimleri arasında tanımlanan kesici yolunu hesaplar, dönüşüm gerçekleştirilmez (Hesaplama kısmı tarafından) 3. Kesici yolu dönüşümü ve kopyası yapılır (Düzeltilme kısmı tarafından) <p>Not: COPY ve TRACUT deyimlerinin ikisi bir kesici yolunun tanımlanmasında kullanıldığında dönüşümler aşağıdaki sırayla uygulanır:</p> <table border="0"> <tr> <td>TRACUT</td> <td>(ilk önce gelen)</td> </tr> <tr> <td>COPY</td> <td>(ikinci önce gelen)</td> </tr> <tr> <td>TRACUT/LAST</td> <td>(üçüncü önce gelen)</td> </tr> </table>	TRACUT	(ilk önce gelen)	COPY	(ikinci önce gelen)	TRACUT/LAST	(üçüncü önce gelen)
TRACUT	(ilk önce gelen)						
COPY	(ikinci önce gelen)						
TRACUT/LAST	(üçüncü önce gelen)						

COPY/1,XYROT,30,11

deyiminin icrasından sonra kesici yolu PTA → 1 → 2 → 3 → 4 → 5 → ... → 1'dir. Kesiciye, K1 etiketli deyimi kullanarak 1 konumundan 2 konumuna yönlendirmeye niyet edilmektedir. Fakat, aşağıdaki sebepten dolayı kesici 1 konumundan W konumuna gider. INDEX/1 ve INDEX/1, NOMORE deyimleri arasında belirlenmiş 1 konumundan başlayarak 2, 3 ve 4 konumlarından geçerek 5 konumuna doğru olan kesici hareketi ile ilgili veriler, hafızada hesaplanır ve saklanır. Her kopyalandığında döndürülürler, fakat son kesici konumu (5. nokta) ve geometrik ögeler (CD ve C1) ile ilgili hafızadaki orjinal veriler değişmez. K1.etiketli deyim icra edildiğinde, kesicinin ilk konumu işlemci tarafından hala KO etiketli deyimle tanımlanan konum (mesela 5. nokta) olduğu farz edilir. Sonuç olarak, kesici 5 noktasından CD

```

ISN 00001 PARTNO TEST MODIFIER LAST
ISN 00002 CLPRNT
ISN 00003 FEDRAT/2.0
ISN 00004 CUTTER/0.5
ISN 00005 C1=CIRCLE/1, 0, 0, 0.2
ISN 00006 REFSYS/(M1=MATRIX/XYROT, 45)
ISN 00007 C2=CIRCLE/1, 0, 0, 0.2
ISN 00008 REFSYS/(M2=MATRIX/XYROT, 90)
ISN 00009 C3=CIRCLE/1, 0, 0, 0.2
ISN 00010 REFSYS/NOMORE; INDEX/1
ISN 00012 FROM/0, 0, 0; TRACUT/(MATRIX/XYROT, 90)
ISN 00014 GOTO/(POINT/CENTER, C1); TRACUT/LAST, (MATRIX/XYROT, 180)
ISN 00016 GOTO/(POINT/CENTER, C2); TRACUT/LAST, NOMORE
ISN 00018 GOTO/(POINT/CENTER, C3); TRACUT/NOMORE; INDEX/1, NOMORE
ISN 00021 COPY/1, TRANSL, 3, 0, 0, 1
ISN 00022 END
ISN 00023 FINI

```

NO DIAGNOSTICS ELICITED DURING TRANSLATION PHASE
17 N/C SOURCE RECORDS (SYSIN)

```

ISN
0001 PARTNO / TEST MODIFIER LAST
0003 FEDRAT/ 2.00000000
0004 CUTTER/ 0.50000000
0011 INDEX/ 1.00000000
0012 FROM /
0.0 0.0 0.0
0013 TRACUT/
0.0 -1.00000000 0.0 0.0
1.00000000 0.0 0.0 0.0
0.0 0.0 1.00000000 0.0

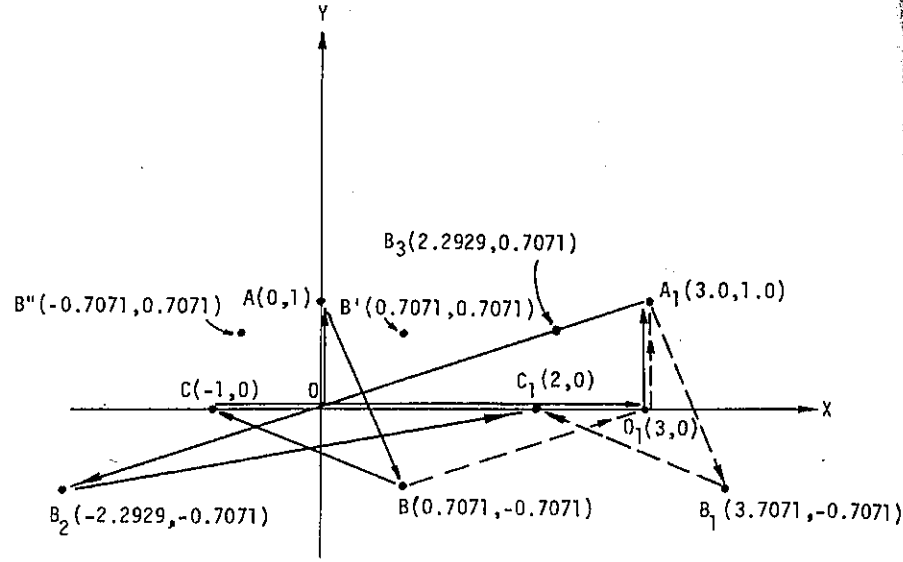
```

```

0014 GOTO/
0.0 1.00000000 0.0
0015 TRACUT/ LAST
-1.00000000 0.0 0.0 0.0
0.0 -1.00000000 0.0 0.0
0.0 0.0 1.00000000 0.0
0016 GOTO/
0.70710678 -0.70710678 0.0
0017 TRACUT / LAST NOMORE
0018 GOTO /
-1.00000000 0.0 0.0
0019 TRACUT/ NOMORE
0020 INDEX 1.00000000 NOMORE
0021 COPY / 1.00000000 TRANSL 1.00000000
3.00000000 0.0 0.0
*** THIS STARTS COPY / 1 PASS 1 ***
0021 FROM /
3.00000000 0.0 0.0
0013 TRACUT/
0.0 -1.00000000 0.0 0.0
1.00000000 0.0 0.0 0.0
0.0 0.0 1.00000000 0.0
0014 GOTO/
3.00000000 1.00000000 0.0
0015 TRACUT/ LAST
-1.00000000 0.0 0.0 0.0
0.0 -1.00000000 0.0 0.0
0.0 0.0 1.00000000 0.0
0016 GOTO/
-2.29289322 -0.70710678 0.0
0017 TRACUT/ LAST NOMORE
0018 GOTO/
2.00000000 0.0 0.0
0019 TRACUT/ NOMORE
0020 INDEX/ 1.00000000 NOMORE
*** THIS COMPLETES COPY / 1 ***
0020 COPY/ 1.00000000 TRANSL 1.00000000
3.00000000 0.0 0.0
0022 END
0023 **** FINI ****
.... END OF SECTION 3....

```

Şekil 7-13 TRACUT deyimleri arasında tanımlanan kesici yolunun dönüşümü ve kopyalanması için bir kopyalama deyimi kullanıldığında TRACUT deyiminde LAST düzelticisinin etkisini gösteren bir bilgisayar programı.



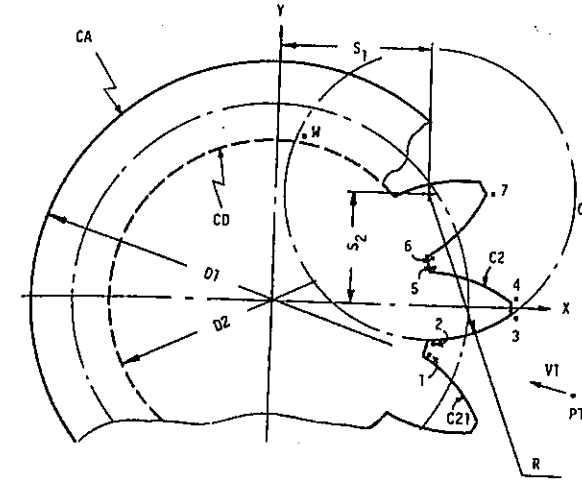
Şekil 7-14 Şekil 7-13'de gösterilen program için kesici yolu

çemberi etrafından C1 denetim yüzeyine ulaşana kadar hareket eder. Nihayet K1 etiketli deyim'in son konumu W noktası olmalıdır.

COPY deyimlerinden hemen sonra artışlı koordinatlarda noktadan noktaya hareketin belirlenmesinde dikkatli olunuz. Başlama konumu, COPY deyimini ile dönüşümü yapılan konum değildir; bu INDEX deyimleri arasındaki son deyimlerle tanımlanan dönüşümü yapılmamış olan konumdur. Onunu için, Şekil 7-15'te listesi verilen programda K etiketli deyim (yani GODLTA/1,0,0,0) için kesicinin başlama konumu 5. noktadır. Eğer 5. noktasının koordinatları (x_5, y_5, z_5) ise son konum $(x_5 + 1, y_5, z_5)$ olacaktır.

7.8 BAŞLATMA VE ÇEVRESEL İŞLEME HAREKETLERİ TANIMLAYAN DİĞER DEYİMLER

Bazen başlatma veya çevresel hareketi, önceki kısımlarda verilen deyimlerle tanımlamak zor olabilir. Böyle durumlarda bu kısımda açıklanan deyimler faydalı olabilir.



Diş sayısı = 12
Diş kalınlığı = 0.40"
 $S_1 = 1.517871032^\circ$
 $S_2 = 1.092365332^\circ$
 $D1 = 4.724409448^\circ$ (Dia.)
 $D2 = 3.149506298^\circ$ (Dia.)
 $R = 1.417322834^\circ$ (Red.)

Program A

```
FROM/PTA
INDIRV/V1
GO/CD, PL1, C21
$$ PL1 IS THE PART SURFACE
AUTOPS
INDEX/1; ARCSLP/ON
TLGRT, GORGT/CD, C1
GORGT/C1, PAST, CA
GOLFT/CA, PAST C2
K0) GOLFT/C2, CD
INDEX/1, NOMORE
COPY/1, XYROT, 30, 11
ARCSLP/ON
K1)TLGRT, GORGT/CD, C1
```

Program B

```
FROM/PTA
INDIRV/V1
GO/CD, PL1, C21
$$ PL1 IS THE PART SURFACE
AUTOPS
INDEX/1; ARCSLP/ON
TLGRT, GORGT/CD, C1
GORGT/C1, PAST, CA
GOLFT/CA, PAST C2
INDEX/1, NOMORE
COPY/1, XYROT, 30, 11
K)GODLTA/1,0,0,0
```

Şekil 7-15 Basitleştirilmiş profilli bir dişli çark. A programı: K1 etiketli deyim niyet edilmemiş bir kesici yolunu belirler. B Programı: COPY deyiminden sonra belirlenmiş GODLTA deyiminin kullanımını açıklar.

7.8.1 Silme-Çıktı Deyimleri (DNTCUT ve CUT)

Bir kesiciyi, şimdiki konumundan doğrudan istenen konuma getirmek zor olduğunda DNTCUT ve CUT deyimleri çifti kullanılır. Şekil 7-16'da verilen örnekte, kesicinin S1 ve S3'e teğet olduğu bir başlatma hareketi, kesiciyi PTA noktasından doğrudan PTB noktasına konumlamak için tanımlanacaktır. Bu, şekilde gösterildiği gibi DNTCUT ve CUT deyimlerini kullanarak gerçekleştirilebilir. DNTCUT deyiminin anlamı, SD işlemciyi, kesici noktalarını (yerleşimleri) sonuçlandıran çıktıları vermeksizin komutları icra etmek için talimat vermektir. CUT deyimini

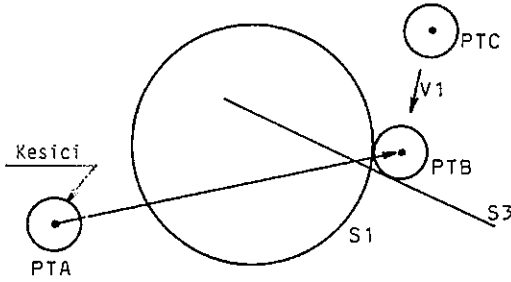
işlemcinin kesici yerleşimlerinin çıktı işlemine yeniden başlayacağını gösterir. CUT komutu belirlendiğinde işlemci tarafından o anki kesici yerleşimi çıktı olarak verilir. Bundan dolayı başlatma hareketi, PTA'dan doğrudan PTB'ye türetilir.

7.8.2 Çoklu Potansiyel Denetim Yüzeyle Çevresel Hareket Deyimi

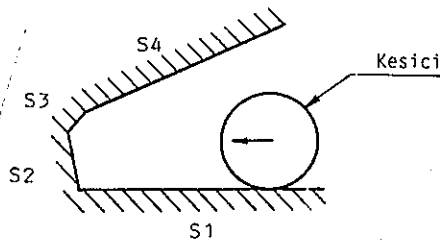
Bazı durumlarda, birkaç yüzeyden herhangi biri, hareket deyiminin denetim yüzeyi olarak alınabilir ve gerçek denetim yüzeyi detaylı hesaplamalar yapılmadıkça tespit edilemez (Şekil 7-17). Bu gibi durumlarda NC işlemcisi önce hangi yüzeyle karşılaşılacağını tespit eder ve sonra kontrolü doğru hareket deyimine aktarır. Bu, bir çoklu denetimli yüzey hareket deyiminin TRANTO deyimi ile birlikte kullanılmasıyla gerçekleştirilir.

APT programı

```
.....
FROM/PTA
DNTCUT
GOTO/PTC
INDIRV/1
GO/S1,S2,S3
CUT
.....
```



Şekil 7-16 DNTCUT ve CUT deyimlerinin kullanımı. S2 parça yüzeyidir (göstermemekte)



Şekil 7-17 Çevresel hareket deyiminin çoklu potansiyel denetim yüzeyleri ile kullanımı için tipik bir durum

Çoklu-denetimli-yüzey hareket deyiminin formatı aşağıda verilmiştir.

$$\left[\begin{array}{l} \{TLFT\} \\ \{TLRGT\} \\ \{TLON\} \end{array} \right] \left\{ \begin{array}{l} \{GOLFT\} \\ \{GORGT\} \\ \{GOFWD\} \\ \{GOBACK\} \\ \{Goup\} \\ \{GODOWN\} \end{array} \right\} / S1, \left\{ \begin{array}{l} \{TO\} \\ \{ON\} \\ \{PAST\} \\ \{TANTO\} \end{array} \right\}, S2, L1, \left\{ \begin{array}{l} \{TO\} \\ \{ON\} \\ \{PAST\} \\ \{TANTO\} \end{array} \right\}, S3, L2, \left\{ \begin{array}{l} \{TO\} \\ \{ON\} \\ \{PAST\} \\ \{TANTO\} \end{array} \right\}, S4, L3, \{f\}$$

Burada:

S1 = Tahrik yüzeyi

S2,S3,S4 = Sırasıyla birinci, ikinci ve üçüncü tahrik yüzeyleri

L1,L2,L3 = Birinci, ikinci ve üçüncü denetim yüzeyleri etkili olduğunda deyimlerin etiketleri sırasıyla dallanacaktır.

f= İlerleme hızı

Bu deyimdeki ilk kelime, önceki deyimdekiyle aynı ise kullanılmayabilir. Deyimlerin ve düzelticilerin karşılık gelen etiketleri ile bu deyimde üçe kadar denetim yüzeyi tanımlanabilir. Çoklu denetimli yüzey deyimi, kontrolü bu muhtemel durumlara karşılık gelen uygun hareket deyimlerine yönlendiren bir şartsız dallanma deyimi olan bir veya birden fazla TRANTO deyimleri ile birlikte kullanılır. Tüm hareket deyimleri APT-AC NC işlemcisinin çeviri kısmı tarafından işlenir ve saklanır; fakat hesaplama kısmında kesici yolu hesaplanana kadar gerçek dallanma geciktirilir.

Şekil 7-17'deki kesici hareketini tanımlamak için çoklu denetimli yüzey hareket deyimini kullanan program parçası aşağıda verilmiştir:

.....
(kesici Şekil 7-17'de gösterilen konumdadır)

TLRGT, GOFWD/S1, S2, K2, S3, K3, S4, K4

\$\$ FORWARD DIRECTION IS A SHOWN IN
\$\$ THE FIGURE

K2) GORGT/S2, S3

K3) GORGT/S3, S4

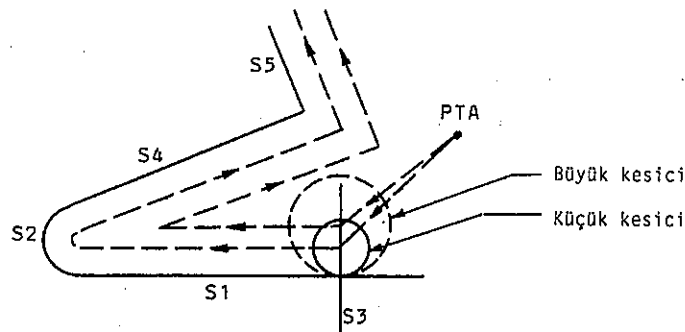
GORGT/S4,

TRANTO/K5 \$\$ THIS STATEMENT DIRECTS CONTROL TO THE ONE LABELED K5

K4) GOBACK/S4,

K5) (Takip eden S4 yüzeyi boyunca tanımlanan hareket deyimi).....

Şekil 7-18 bu deyimlerin kullanımının bir örneğini gösterir. Bu tezgah işleminde, aynı profil, farklı ölçüdeki iki kesici ile (ölçüsü daha büyük kaba talaş kesici ve ölçüsü küçük bitirme kalemi) işlenecektir. S2 olarak tanımlanan kavis, programda çoklu denetimli yüzeylerin ve kontrolün dallanmasının belirtilmesini gerektirir. Küçük ve büyük kesiciler ve kesicilerin ikisini de yönlendirmek için APT program parçaları verilmiştir. İki kesici için olan program parçasına THICK deyimi ile birlikte bir MACRO deyimi eklenebilir. Sonra bu MACRO (alt program), kaba ve ince işlemler tanımlamak için çağrılabilir.



Küçük kesici için APT deyimleri:

```
FROM/PTA
GO/ON,S3,PS1,S1
TLRGT,GORGT/S1,TANTO,S2
GOFWD/S2,TANTO,S4
GOFWD/S4,PAST,S4
```

Büyük kesici için APT deyimleri:

```
FROM/PTA
GO/ON,S3,PS1,S1
TLRGT,GORGT/S1,S4
GORGT/S4,PAST,S5
```

Büyük ve küçük kesiciler için APT deyimleri

```
FROM/PTA
GO/ON,S3,PS1,S1
TLRGT,GORGT/S1,TANTO,S2,$
ID1,TO,S4,ID2
ID1)GOFWD/S2,TANTO,S4
GOFWD/S4,PAST,S5
TRANTO/ID3
ID2)GOBACK/S4,PAST,S5
ID3)GOLFT/S5,.....
```

Şekil 7-18 Çoklu denetimli yüzey deyimi kullanımı örneği. PS1 parça yüzeyi (göstermemiş) kesici eksenine diktir.

7.8.3 Tahrik Yüzeyinin Denetim Yüzeyi ile Seçilmiş Kesişimini Belirten Hareket Deyimi

Bir tahrik yüzeyi, denetim yüzeyi ile birkaç arakesisi olduğunda, 5. Bölümde açıklanan çevresel hareket deyimi kullanılırsa kesici ilk kesişimde hareket yönünü değiştirecektir. Daha sonra birkaç çevresel hareket deyimleri, ilk kesişmeden n'inciye kadar aynı tahrik yüzeyi boyunca kesici hareketinin tanımlanmasına gerek vardır. Bu durumda programlamayı basitleştirmek için bir parametre ile birlikte ek bir kelime, INTOF çevresel harekette istenilen son konumu göstermek için kulla-

ılabilir. Kesici hareketinin yönünü değiştirmek için istenen konumun denetim yüzeyi ile tahrik yüzeyinin n'inci kesişmesi olduğunu kabul edelim; "n, INTOF" seçeneği için tahrik yüzeyinin önünde hareket deyimi içinde sunulabilir. Formatı aşağıdaki gibidir:

$$\left[\begin{array}{l} \text{TLRGT} \\ \text{TLRGT} \\ \text{TLON} \end{array} \right], \left\{ \begin{array}{l} \text{GOLFT} \\ \text{GORGT} \\ \text{Goup} \\ \text{GODOWN} \\ \text{GOFWD} \\ \text{GOBACK} \end{array} \right\} / S1, \left[\begin{array}{l} \text{TO} \\ \text{ON} \\ \text{PAST} \\ \text{TANTO} \end{array} \right], n, \text{INTOF}, S2 [, f]$$

burada S1 tahrik yüzeyi; S2, denetim yüzeyi ve f ilerleme hızıdır. n parametresi pozitif bir tamsayı olmalıdır. Şekil 7-19'da bir örnek verilmektedir.

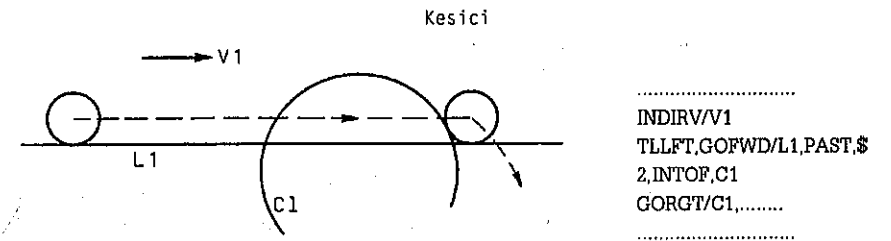
7.8.4 OFFSET Deyimi

OFFSET deyimi, başlatma hareketinde son konumu kontrol için ek vasıta temin etmek için kullanılır. Bu deyimin formatı:

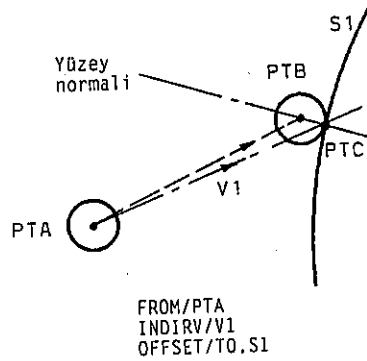
$$\text{OFFSET} / \left[\begin{array}{l} \text{TO} \\ \text{ON} \\ \text{PAST} \end{array} \right], S1 \left[\begin{array}{l} \text{TO} \\ \text{ON} \\ \text{PAST} \end{array} \right], S2 [, f]$$

Bu deyimdeki S1 ilk yüzey, tahrik yüzeyidir ve S2, parça yüzeyidir. Deyimin bu parçasındaki köşeli parantezler, her zaman olduğu gibi gerekli değilse kullanılmayabilir.

OFFSET deyimi INDIRV veya INDIRP deyiminden önce gelmelidir. INDIRV veya INDIRP deyimi ile tanımlanan vektör o anda geçerli olan kesici konumuna (PTA) bağlanacağı düşünülür; uzatılınca OFFSET deyiminde tanımlanan tahrik yüzeyi (S1) ile kesişir (Şekil 7-20). O noktadan tahrik yüzeyinin normali



Şekil 7-19 Denetim yüzeyi ile tahrik yüzeyinin seçilen kesişimini tanımlayan hareket deyimi.



Şekil 7-20 OFFSET deyimini tanımlayan başlatma hareketi.

FROM/PTA
INDIRV/V1
OFFSET/TO.S1

SD işlemcisi tarafından bulunur ve başlatma hareketinin denetim yüzeyi olarak farzedilir. Böylece, kesici tahrik yüzeyinin normali üzerinde tanımlanan düzelticiye (Şekil 7-20'deki düzeltici TO'dur) karşılık gelen geçerli bir konum olan PTA'dan doğrudan PTB'ye hareket edecektir. Eğer parça yüzeyi, OFFSET deyiminde tanımlanırsa kesici, kesici eksenini boyunca TO, ON veya PAST ile parça yüzeyine hareket edecektir (TO=parça yüzeyine doğru, ON=üzerine, PAST=yüzeyi geç); aksi takdirde önceden tanımlanan parça yüzeyine hareket edecektir. PTA'dan PTB'ye ve PTB'den parça yüzeyine olan bu iki hareket aynı anda bitirilecektir; böylece işlemci, PTA'dan PTD'ye olan sadece bir hareket deyimini türetir.

7.9 POKKED (CEP) DEYİMİ

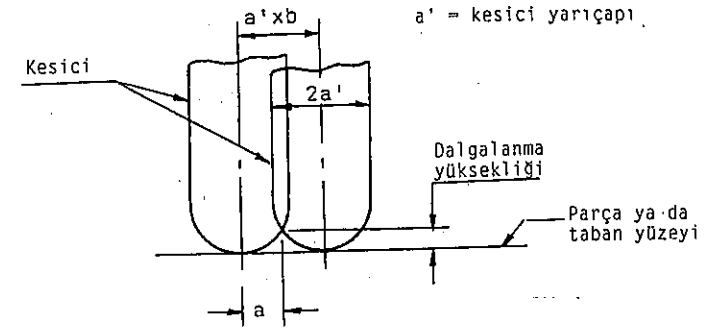
Bazen istenilen bir derinlikte bir seri doğru parçaları ile çevrelenmiş bir bölgeyi işlemeye gerek duyabiliriz. POKKED deyimini, birçok deyim ve hesaplamadan kaçınılmak istendiği durumda kullanılabilir.

POKKED deyiminin formatı aşağıdaki gibidir:

POKKED/a,b,c,f1,f2,f3,d,e,PT1,PT2,...PTn

burada

- a = Kesici geometrisi tarafından tespit edilen etkili kesici yarıçapı, cebin taban yüzeyi ve istenilen dalgalanma yüksekliği (Şekil 7-21).
- b = Bir işlemenden diğer bir işlemeye olan kademe ilerleme etkeni (kademe ilerlemesi veya kaydırma b ile çarpılan kesici yarıçapı a'dır ve daima parça [taban] yüzeyinde ölçülür.



Şekil 7-21 POKKED deyiminde kullanılan a ve b parametreleri.

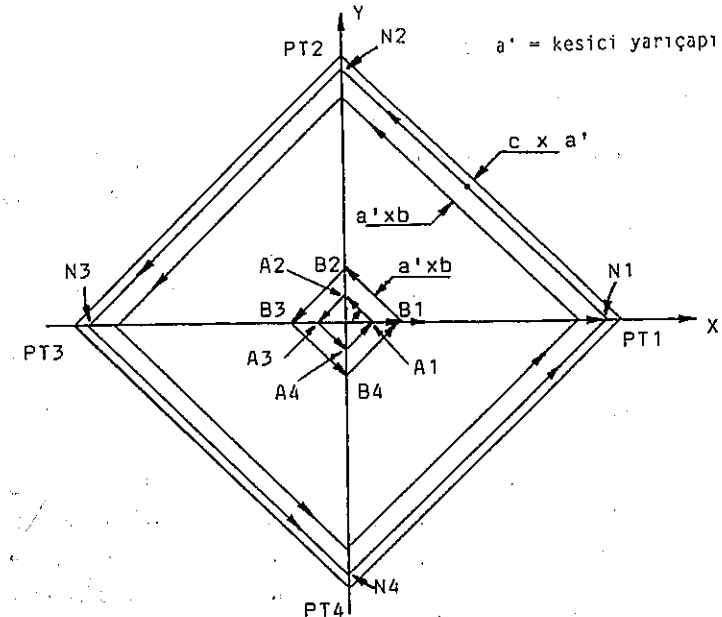
c = Bitirme işlemi için kademe ilerleme faktörü (bitirme işlemi için kademe ilerleme kesici yarıçapı ile, a', c'nin çarpımına eşittir. Benzer şekilde bu da parça yüzeyinde ölçülür).

f1 = Kesicinin cebe olan ilk hareketi için ilerleme.

f2 = Genel cep işlemesi için ilerleme.

f3 = Cep çevresinde bitirme işlemi için ilerleme.

d = 0 (İşlemci, dalgalanma yüksekliğinin izin verilen değerden az olup olmadığından emin olmak için test edecektir. Eğer değilse, yeni kademe ilerleme oranları tanımlanan b ve c'yi değiştirmek için



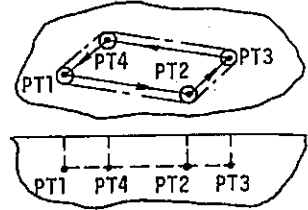
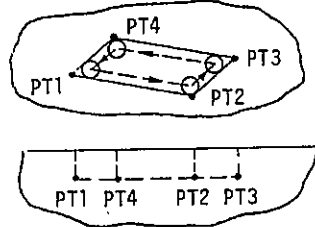
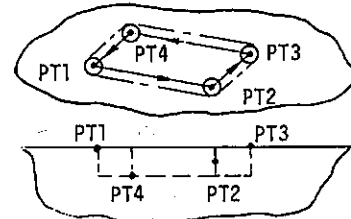
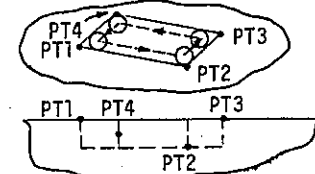
Şekil 7-22 Şekil 7-23'te gösterilen cep işleme programı kesici yolu.

işlemci tarafından ayarlanacaktır).

$d = 1$ (İşlemci tarafından test yapılmaz. Cep işleme boyunca b ve c programlanmış kademe ilerlemeleri kullanılacaktır).

$PT1, PT2, \dots, PTn$ = İşlenecek cebin sınır köşelerini tanımlayan noktalar.

e = $PT1 - PTn$ arasındaki noktaların tiplerini gösteren parametre ve taban yüzeyinin biçimi (aşağıdaki tabloya bakınız).

e	CEBİN TABANI	CEBİN KÖŞESİ
0		Kesici merkez noktası ile geçilen girdi noktaları
		
1	$PT1 - PTn$ arasındaki girdi noktaları cebin tabanı olan düzlem üzerindedir.	Girdi noktaları cebin köşeleridir.
		
2	Parça yüzeyi cebin tabanı olarak kullanılır; $PT1 - PTn$ arasındaki girdi noktalarının bir düzlemde olması gerekmez.	Kesici-merkez noktası ile geçilen girdi noktaları
		
3		Girdi noktaları cebin köşeleridir.
		

Girdi noktalarının sırası, cep temizlenirken kesicinin gideceği yönü tespit eder.

POCKET deyimi hareket deyimidir. Bir kere icra edilince kesici cebin merkezine yakın (z koordinatı cebin tabanı ile aynı olan) bir başlama noktasına gidecek ve cebi temizleyecektir. Başlama noktası işlemci tarafından aşağıdaki yolla tespit edilir (Şekil 7-22). NC işlemcisi önce bitirme işlemi için kesici konumlarını (yani $PT1, PT2, PT3$ ve $PT4$ noktalarını) hesaplar ve c faktörüne göre son kaba işleme noktalarını ($N1, N2, N3$ ve $N4$) hesaplamak için geri döner. Geri kalan kesici konumları, a' ve b faktörleri ile birincisi son kesici konumu ve sonuncusu ilk kesici konuma dayalı olarak hesaplanır. Son olarak $A1-A2-A3-A4$ ilk hareket çevrimindeki ilk nokta (yani $A1$ noktası) cep işleme için başlama noktası olarak seçilir.

POCKET deyimi için kesici hareketi şöyledir: Kesici o andaki konumundan doğrudan cepteki başlama noktasına hareket eder ve sonra cebin sınırlarına doğru yavaş yavaş hareket ederek girdi noktalarının sırası ile tanımlanan yönde spiral hareketi yapar. Örneğin, Şekil 7-23'te verilen programdaki POCKET deyimi kesiciyi $A1$ konumuna getirecektir (Şekil 7-22). Sonra kesici $A1-A2-A3-A4-A1 \dots N1-N2-N3-N4-N1-PT1-PT2-PT3-PT4-PT1$ yolu boyunca hareket edecektir. Kesici $PT1$ 'de tekrar yukarı hareket edecektir.

```

ISN 00001 PARTNO TEST POCKET
ISN 00002 CLPRNT
ISN 00003 CUTTER/0.2
ISN 00004 PT1=POINT/1, 0, 0
ISN 00005 PT2=POINT/0, 1, 0
ISN 00006 PT3=POINT/-1, 0, 0
ISN 00007 PT4=POINT/0, -1, 0
ISN 00008 FROM/0, 0, 0.1
ISN 00009 RAPID
ISN 00010 GOTO/0, 0, 0
ISN 00011 PSIS/(PLANE/0,0,1,-0.5)
ISN 00012 POCKET/0.1, 0.9, 0.1, 1.0, 2.0, 1.5, 0, 2, PT1, PT2, PT3, PT4
ISN 00013 GODLTA/1.5
ISN 00014 END
ISN 00015 FINI
NO DIAGNOSTIC ELICITED DURING TRANSLATION PHASE
15 N/C SOURCE RECORDS (SYSIN)

```

```

ISN
0001 PARTNO/TEST POCKET
0003 CUTTER/ 0.200000000
0008 FROM /
0.0 0.0 0.10000000
0009 RAPID
0010 GOTO/
0.0 0.0 0.0

```

```

0012
0012 FEDRAT/ 1.00000000
0012 GOTO / POCKET ( )
0.09490332 0.0 -0.50000000

0012 FEDRAT/ 2.00000000
0012 GOTO / POCKET ( )
0.0 0.0940332 -0.50000000
-0.09490332 0.0 -0.50000000
0.0 -0.09490332 -0.50000000
0.09490332 0.0 -0.50000000
0.22218254 0.0 -0.50000000
0.0 0.22218254 -0.50000000
-0.22218254 0.0 -0.50000000
0.0 -0.22218254 -0.50000000
0.22218254 0.0 -0.50000000
0.34946176 0.0 -0.50000000
0.0 0.34946176 -0.50000000
-0.34946176 0.0 -0.50000000
0.0 -0.34946176 -0.50000000
0.34946176 0.0 -0.50000000
0.47674098 0.0 -0.50000000
0.0 0.47674098 -0.50000000
-0.47674098 0.0 -0.50000000
0.0 -0.47674098 -0.50000000
0.47674098 0.0 -0.50000000
0.60402020 0.0 -0.50000000
0.0 0.60402020 -0.50000000
-0.60402020 0.0 -0.50000000
0.0 -0.60402020 -0.50000000
0.60402020 0.0 -0.50000000
0.73129942 0.0 -0.50000000
0.0 0.73129942 -0.50000000
-0.73129942 0.0 -0.50000000
0.0 -0.73129942 -0.50000000
0.73129942 0.0 -0.50000000
0.85857864 0.0 -0.50000000
0.0 0.85857864 -0.50000000
-0.85857864 0.0 -0.50000000
0.0 -0.85857864 -0.50000000
0.85857864 0.0 -0.50000000
0.98585786 0.0 -0.50000000
0.0 0.98585786 -0.50000000
-0.98585786 0.0 -0.50000000
0.0 -0.98585786 -0.50000000
0.98585786 0.0 -0.50000000

```

```

0012 FEDRAT/ 1.50000000
0012 GOTO/ POCKET ( )
1.00000000 0.0 -0.50000000
0.0 1.00000000 -0.50000000
-1.00000000 0.0 -0.50000000
0.0 -1.00000000 -0.50000000
1.00000000 0.0 -0.50000000
0013 GOTO/ TLAXIS
1.00000000 0.0 1.00000000

0014 END
0015 ***** FINI *****
.... END OF SECTION 3 ....

```

Şekil 7-23 Bir cep işleme programı ve onun CLDATA'sı.

POCKET deyimindeki girdi noktaları ile tanımlanan poligonun APT-AC NC işlemcisi için tümsel olması gerekir. Bu, poligonun iç köşesindeki açının 180 dereceden az olması gerektiği anlamına gelir. Şekil 7-23'te gösterilen APT programında, PT1 noktasındaki iç açının 180 dereceden büyük olduğu anlamına gelen PT1 noktası

PT1=POINT/-1.0,0,0

olarak tanımlanırsa (2504 nolu) hata mesajı

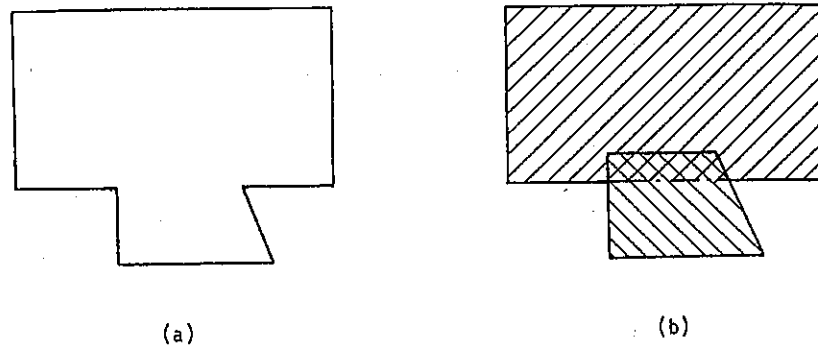
"POCKED INPUT POINTS DO NOT SPECIFY A CONVEX POLYGON" verilir.

Bitişik kenarla arasında 180 dereceden fazla iç açılı bir girinti yalnızca tümsel olan ceplerin kenarları tarafından biçimlenen çokgenli iki veya daha fazla ceplere bölünmelidir. Girintinin tamamen temizlendiğinden emin olmak için işlenen bölgede üst üste binmeler de gerekir (Şekil 7-24).

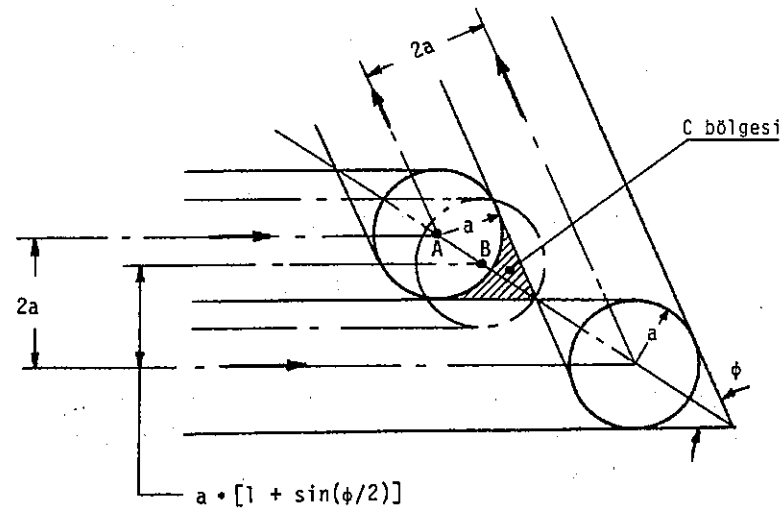
90 dereceden daha az bir iç açılı girinti için POCKET deyimini kullanıldığında köşede işlenmemiş bölge kalmadığından emin olmak için b kademe ilerleme faktörü ayarlanmalıdır. Kademe ilerlemesi, $b \cdot a'$, $2 \cdot a'$ dan daha az olmalı ve aşağıdaki formül ile hesaplanabilir (Şekil 7-25).

$$b \cdot a' \leq a \cdot [1 + \sin(\phi/2)]$$

NC işlemcisi için, tanımlanan cebin taban yüzeyi ile kesici eksenindeki açının 1 derece 10 dakikadan daha büyük olması gerekir.



Şekil 7-24 İç açılmanın 180 dereceden büyük olduğu bir cep (b) birbirinin üstüne binen (a) iki cebe bölünmelidir.



Şekil 7-25 90 dereceden az bir iç açılı bir cep için kademe ilerlemesinin tespiti. Eğer kademe ilerlemesi etkin yarıçapın (yarı $2 \cdot a$) iki katı ise C bölgesi işlenmemiş olarak bırakılır. Kademe ilerlemesi, $b \cdot a'$,

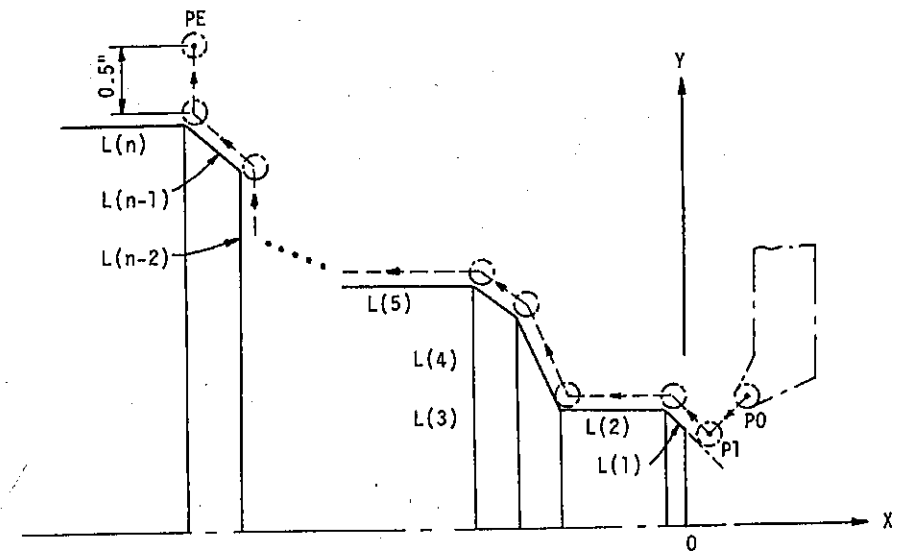
$$a \cdot [1 + \sin(\phi/2)]$$

indirgenmeli veya bölgeyi temizlemek için kesicinin merkezi A noktasından B noktasına hareket ettirilmelidir.

Doğrusal olmayan sınırı olan bir cep, sınırın üstünde veya içinde seçilen birkaç nokta ile bir çokgen olarak basitleştirilmelidir. Çokgen cebinin işlenmesinden sonra eğri sınır boyunca bir çevresel hareket gerekir.

PROBLEMLER

- 7.1 Genellikle bir frezeleme işlemi iki kısma bölünür; biri kaba işleme ve diğeri de ince işleme. Bu frezeleme işlemi tanımlamak için MACRO (alt program) kullanan bir APT programının genel yapısını veriniz.
- 7.2 AVOID düzelticili bir PATTERN deyimi ile tanımlanan bir delme hareketi sadece bir hareket hızına sahip olabilir. Fakat, her deliğin delinmesi için üç hareket kademesi iki farklı hız da olmalıdır; bunlar şunlardır:
- Delğin üst yüzeyinin hemen üstündeki konuma hızlı hareket
 - Delğin derinliğine tanımlanan ilerleme hızında aşağı doğru bir hareket.
 - Gerekli seviyeye kadar yukarı doğru hızlı geri çekilme hareketi.
- Eğer bir parçadan n adet deliğin aynı $H1$ derinliğine, sahip olduğu $n+m$ adet delik varsa, diğer m adet delik $H2$ derinliğine sahiptir. " i "nci deliğin x ve y koordinatları $x(i)$ ve $y(i)$ 'dir. Bu deliklerin üst yüzeyleri aynı seviyededir: $z=z1$. Delik delme ilerleme hızı $f1$ 'dir. Diğer işleme özellikleri rastgele atanabilir. Bu delikleri delmek için mümkün olduğu kadar kısa ve açık bir APT programı yazınız.
- 7.3 Sık sık tornalamanız gereken millerden biri Şekil P7-3'te gösterilmektedir. Bunun temel özellikleri şunlardır:
- Profil sadece doğrulardan oluşmaktadır.
 - Çapı- x yönünde yeknesak bir şekilde artmaktadır.
- Koordinat sisteminin orijininin daima uç yüzeyin merkez noktasında ayarlandığını kabul ediniz. Ayrıca, ilk doğru parçası uç yüzeyden sonra gelendir;



Şekil P7-3

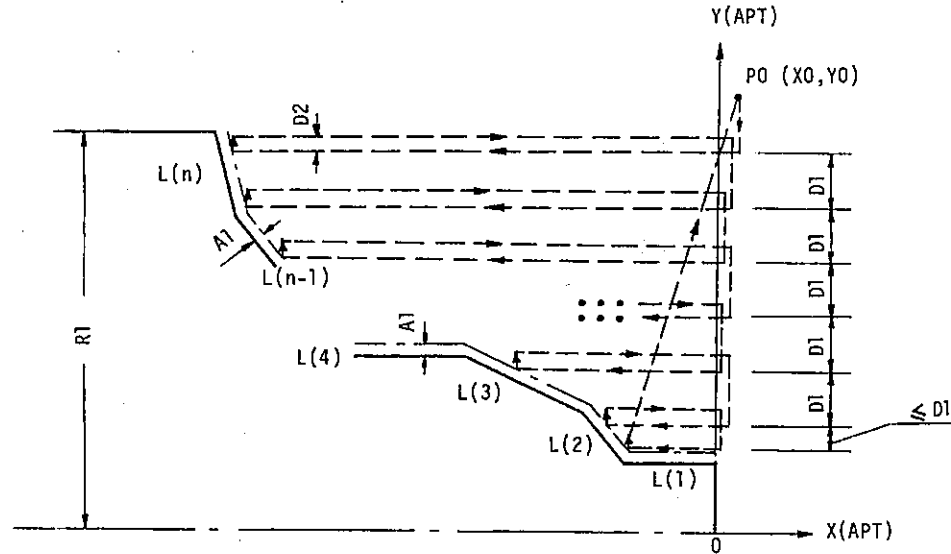
yani L(1), 0 ile -90 derece arasında bir eğik açılı doğru olabilir. Bitirme işlemi için kesicinin başlama konumu, P0, daima başlatma hareketinin son konumunun (yani P1) uç yüzeyin sağ kenarında olacak şekilde seçilir. Profili istenen herhangi bir sayıdaki doğru parçalarından oluşan parçalar için otomatik olarak bitirme kesici yolunu, P0-P1-.....-PE, tanımlamak için çağrılabilen bir APT programını veya alt programlarını yazınız.

7.4 Şekil P7-4'te gösterilen parça Problem 7.3'te açıklama ile aynıdır. Onun kaba tornalama işlemi bir kaç hareket çevriminden ibarettir. Bu kaba işlemi tanımlamak için mümkün olduğu kadar kısa ve açık bir APT Programı yazınız. Kabul edilen şartlar aşağıda verilmiştir:

- Parça profili sadece doğru parçalarından ibarettir. İlk parça, L(1) X eksenine paralel ve sonucu, L(n) değildir.
- Her paso için talaş derinliği D1'dir.
- Geri çekilme hareket mesafesi D2'dir.
- İzin verilen pay A1'dir.
- Başlama ve bitiş noktaları (X0, Y0)'dır.
- İlerleme hızı 0,007 inç/devir ve devir sayısı 500 dev/dak.
- Başlama noktası, P0, her zaman şöyle seçilir. $Y0 - D1 < R1$.

7.5 Şu deyimle tanımlanan dönüşümlerin sırasını bir örnek vererek doğrulayınız.

MATRIX/M1, TIMES, M2



Şekil P7-4

- 7.6 Tablo 7-3'te verilen MATRIX deyiminin 9 no.lu formatı, yerel koordinat sisteminde tanımlanan bir geometrik öğeyi taban koordinat sisteminde tanımlanan öğeye dönüşümünü yapmak için kullanıldığını gösteren bir örnek veriniz.
- 7.7 Şekil 7-15'te gösterilen dişli çark herbiri 0,06 inçlik bir talaş derinliği olan 7 pasoda, $\frac{1}{8}$ inç çaplı bir düz uçlu parmak freze ile işlenecektir. Kesici yolunun tanımlanması ve dönüşümü için sadece COPY dönüşüm deyimini kullanan bir APT programı yazınız. Kabul edilmiş şartlar aşağıda verilmiştir:
- Başlama ve bitiş noktaları dişli çarkın merkezindedir ve yüzeyden 1 inç yukarıdadır.
 - Devir sayısı = 1000 dev/dak.
 - İlerleme = 2 inç/dak.
- 7.8 Geometrik öğelerin dönüşümü için sadece REFSYS dönüşüm deyimini kullanarak problem 7.7'yi tekrar cevaplayınız ve daha sonra kesici yolunu tanımlayınız.
- 7.9 Aşağıda listesi verilen program, $z = -0,06$ inç seviyesinde, problem 7.7'de verilen dişli profilini işlemek için kesici yolunu tanımlamak maksadı ile TRACUT deyimini kullanmaktadır. Önce kesici yolunun, 3 noktasından 4, 5 ve 6 noktaları aracılığı ile 7 noktasına gidişi tanımlanır. Herbiri artışı 30 derecelik dönme açısı ile kesici yolu 11 kere döndürülür. Aşağıdaki program, dişli profilini işlemek için doğru kesici yolunu tanımlamakta mıdır? Eğer değilse, sebebini açıklayınız ve programı düzeltiniz.

```

PARTNO PROBLEM 7.9; THE ORIGIN OF THE COORDINATE SYSTEM IS ON
$$ TOP SURFACE
MACHIN / .....
CLPRNT
RESERV/M1, 12
C1=CIRCLE/.....
C21=CIRCLE/.....
CD=CIRCLE/.....
CA=CIRCLE/.....
C2=CIRCLE/.....
P1=POINT/(2.2*COSF (-15)), (2.2*SINF (-15)), 1 $$$ A POINT WITHIN THE
$$$ AREA BOUNDED BY CIRCLE C1, CD, C21, AND CA
DO/K1, I=1, 12
K1) M1 (1)=MATRIX/XYROT, (30*(I-1))
REFSYS/M1 (2)
C3=CIRCLE/.. (C1'i tanımlamak için kullanılan parametrelerin aynısı)...
REFSYS/NOMORE
CUTTER/ (1/8)
SPINDL/1000, CLW

```



```

COOLNT/ON
FROM/0, 0, 1.0, 2.0
RAPID; GOTO/P1
GODLTA/-1.06
GO/C1, (PLANE/0, 0, 1, -0.06), PAST, CA
DO/K2, J=1, 12
  TRACUT/M1(J)
    ARCSLP/ON; TLRGET, GOLFT/CA, PAST, C2
    GOLFT/C2, CD
    GORGT/CD, C3
    GORGT/C3, PAST, CA
  TRACUT/NOMORE
K2) CONTIN
RAPID; GODLTA/1.06
RAPID; GOTO/0, 0, 1.0
SPINDL/OFF
COOLNT/OFF
END
FINI

```

```

TRACUT/NOMORE
K2) CONTIN
RAPID; GODLTA/1.06
RAPID; GOTO/0, 0, 1.0
SPINDL/OFF
COOLNT/OFF
END
FINI

```

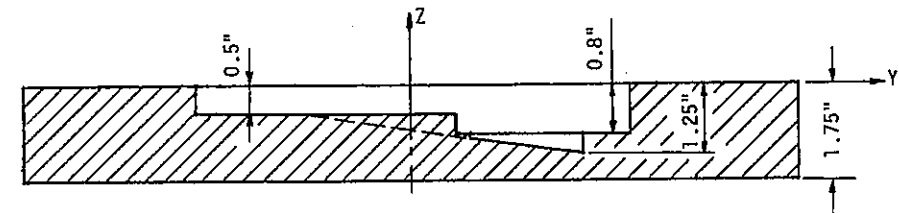
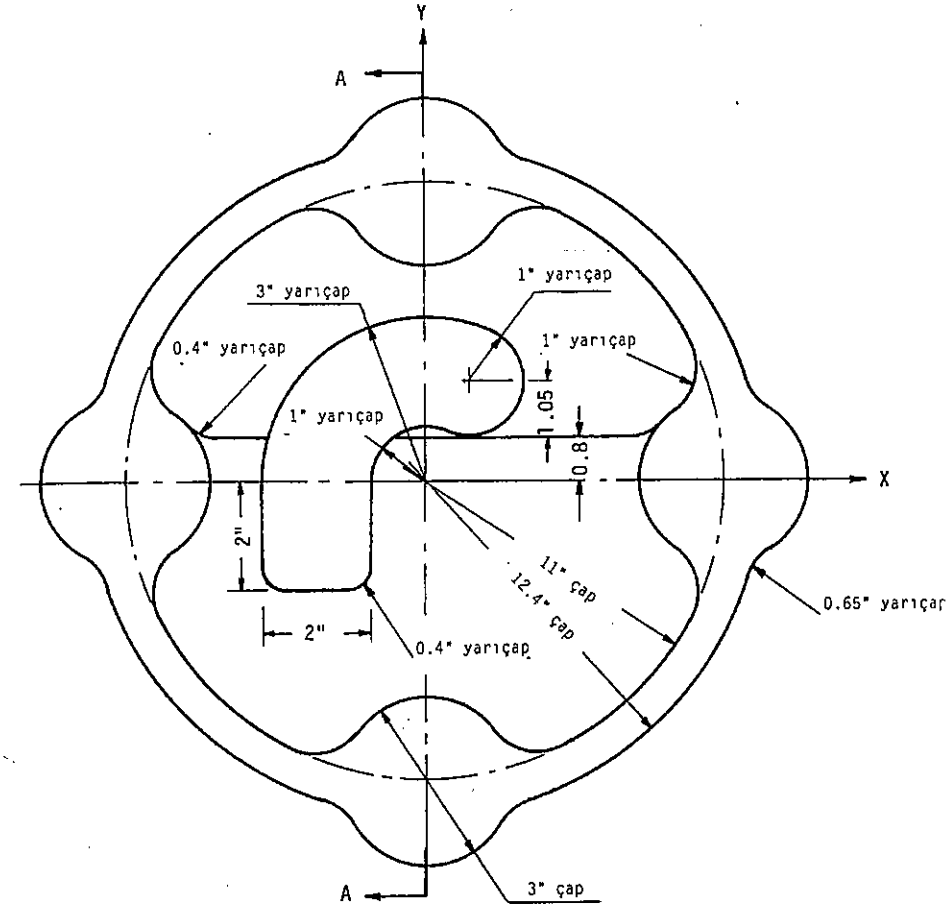
7.10 Şekil P7-10'da gösterilen parça dökme demirden yapılmıştır. Dış profili, döküm işlemi ile biçimlendirilmiştir. İç profil, birisinin açılı taban yüzeyine sahip olduğu üç girintiye (cebe) bölünebilir. Üst yüzeyin istenen yüzey kalitesi ve ölçüde işlenmiş olduğunu farz ediniz. İç profili işlemek için bir APT programı yazınız. Aşağıdaki durumları kabul ediniz:

- Kesici: $\frac{3}{4}$ inç çaplı düz uçlu bir parmak freze.
- Devir sayısı: 500 dev/dak.
- İlerleme hızı: 3 inç/dak. (kaba işleme)
2 inç/dak. (inç işleme)
1 inç/dak. (dalma)
- Soğutucuya gerek yoktur.
- İnce işleme payı: 0.02 inç.
- Başlama ve bitiş noktası: (10,5,3)
- İzin verilen dalgalanma yüksekliği: 0.005 inç.

Not: Düz uçlu parmak freze vasıtası ile yatay olmayan bir taban yüzeyli cebi işlemek için kademe ilerlemesi, f, şu ifadeler kullanılarak tespit edilebilir:

$$F = (2H/\sin A) \cdot [R \cdot (2 \sin A/H - 1)^2] \quad (0 < A < 90^\circ)$$

$$F \leq R$$



A-A görünüşü

Şekil P7-10

burada

H = İzin verilebilir dalgalanma yüksekliği

R = Kesici yarıçapı

A = Taban düzlemi ile yatay düzlem arasındaki açı.

Bölüm 8

Bir APT Programının Yapısı ve İşlenmesi

3. Bölümden 7. Bölüme kadar bir işleme programı oluşturmak için gerekli deyimleri detaylı olarak takdim edip açıkladık. Şimdi, doğru bir APT parça işleme programı oluşturulması ve bunun nasıl işlenip hatalarının bulunması ile gerekli deyimlerin nasıl biraraya konacağı sorusu ortaya çıkmaktadır. Böylece bu bölümde, bir APT programının yapısını ve genel olarak bir programcının program yazımı hakkında ne bilmesi gerektiğini tartışacağız. Tartışma, torna ve freze işlemlerinin programlanması arasındaki farkla başlamakta; bunu bir son işlemci ve APT deyimleri arasındaki ilişki takip etmektedir. Sonra bir APT programının genel yapısı tartışılmaktadır. Bir APT programının nasıl oluşturulduğunu açıklamak için dört örnek verilmektedir.

Bir APT programı APT-AC NC işlemci tarafından işlenmelidir ve NC'li tezgahı için özel olarak tasarlanan son işlemci kullanılmalıdır. Programcının bir programdaki söz diziminde, programlama mantığına veya ikisinde birden hata yapması kaçınılmazdır. Genellikle, NC işlemci ve son işlemci tarafından verilen hata mesajlarına göre hataları düzeltmek için programı birkaç kere çalıştırılmalıdır (programı işlemek için NC işlemciyi ve son işlemciyi kullanılmalıdır). Hata ayıklama olarak bilinen bu işlem, yeni bir programcı için programı yazmaktan daha uzun zaman alabilir. Hatta tecrübeli bir APT programcısı dahi bir APT programını birkaç kere çalıştırma ihtiyacı hissedebilir. Onun için, doğru ve faydalı bir program üretmek için bir programı çalıştırmak ve hatasını ayıklamak gerekir.

Bu bölümün ikinci kısmında, önce APT-AC NC işlemci ile bir programın işlenmesini sunacağız ve bir APT programını çalıştırmak için detaylı olarak yön-

temleri ve gerekli iş icra tarafından verilen hata mesajlarını tartışacağız. Son olarak Bölüm 8.6'daki tartışmalar IBM VM/CMS (Virtual Machine / Conversational Monitor System) işletim sistemine dayalıdır. Bir APT programını IBM OS/VS (Operating System/Multiple Virtual System) ortamında çalıştırmak için yöntemler benzerdir ve Ek D'de gerekli iş kontrol dili (JCL) deyimleri listelenmiştir.

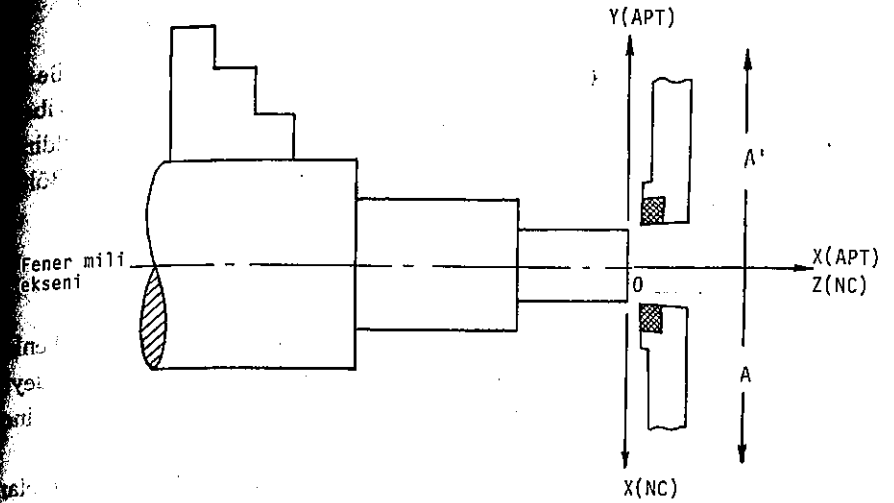
8.1. TORNALAMA İŞLEMLERİNİN PROGRAMLANMASI İÇİN KOORDİNAT SİSTEMİ

Önceki bölümlerde görüldüğü gibi, tartışmalarımız genellikle frezeleme işlemleriyle ilgilidir. APT dili, X-Y düzleminde iki boyutlu durumları ele alan birçok deyim sağlar. Bir tornadaki tornalama işlemi de iki boyutludur; fakat X-Z düzleminde icra edilir. Gerçekte, bir torna kalem, torna kalemünün uç yarıçapına eşit olan ve eksen Z yönünde olan bir düz uçlu parmak freze olarak APT'de işlem görür. Böylece, bir tornalama işleminin kesici yolu APT programında X-Z düzlemi yerine X-Y düzleminde tanımlanmalıdır (Şekil 6-11 [a]). Torna için son işlem, genellikle APT koordinat sisteminde x ve y koordinatları, NC çıktısında veya programda z ve x koordinatlarına aktarılacak şekilde tasarlanır. Şekil 6-11 (a)'da gösterilen torna tezgahı için hala bir problem vardır: Kesici APT koordinat sistemi Y ekseninin negatif tarafındadır ve o zaman (NC'li tezgah koordinat sisteminin) X yönündeki NC çıktısı negatif olur ve bu doğru değildir.

Bu problemi çözmek için iki yol vardır: (1) APT programındaki pozitif y koordinatı NC programındaki negatif x koordinatına değiştirecek ve (2) kesici yolunun gerçek konumunda tanımlayacak (yani APT koordinat sisteminin Y-negatif tarafında) bir yolla son işlemciyi tasarlamaktadır. Fakat, bütün y koordinatları APT programında negatif olacağından bu uygun değildir. Diğer bir yol, bir tormalama işlemi için, kesiciyi iş mili ekseninin herhangi bir tarafında yerleştirmekle ortaya çıkan parça profilinde bir farklılık oluşmadığından dolayı, kesici yolunu APT koordinat sistemindeki y ekseninin pozitif tarafında (Şekil 8-1'deki A tarafı) tanımlamaktır. Fakat, fener milinin dönüş yönü ve kullanılan kesici tipleri gerçek kesici konumuna (Şekil 8-1'de gösterilen durum için fener mili ekseninin A tarafına) dayalı olarak tespit edilmelidir. Bir çok torna ve tormalama merkezleri, kesicilerin fener mili ekseninin üst tarafına, A (Şekil 8-1) yerleştirilecek şekilde tasarlanmaktadır. Bu durumda, APT programındaki kesici konumunu değiştirmeye gerek yoktur.

8.2 SON İŞLEMCİNİN APT PROGRAMLANMASINA ETKİSİ

1. Bölümde belirtildiği gibi, bir APT programı hem APT-AC NC işlemcisi hem de seçilen bir son işlemci tarafından işlenir. Onun için deyimlerin formatları



Şekil 8-1 APT ve NC koordinat sistemleri arasındaki fark. Gerçekte torna kalem iş milinin aşağı A tarafındadır. APT programlama esnasında yukarı tarafta, A' olduğu farz edilebilir.

ve kullanımları bu iki işlemcinin tasarımı tarafından tespit edilir. Geometrik tanımlamalar, kesici hareketleri, işleme özellikleri ve bilgisayar işlemlerini de içine alan APT deyimlerinin çoğu APT-AC NC işlemcinin kendisi tarafından tespit edilir. Fakat, formatları hem NC işlemci hem de seçilen son işlemci tarafından tespit edilen bir çok deyim vardır. Örneğin, eğer bir freze tezgahı için sadece iki soğutucu durumu (ON=açık ve OFF=kapalı) varsa, son işlemci, COOLNT deyiminde ON ve OFF'tan başka kelimeleri kabul etmeyecek şekilde tasarlanabilir; böylece FLOOD, MIST ve TAPKUL (Bölüm 7.6'ya bakınız) kelimeleri eklenebilir. Benzer şekilde, devir sayısı ile ilgili (Bölüm 6.5'e bakınız) S parametresi, tezgah veya son işlemcinin tasarımına bağlı olarak en çok verilebilen hız yüzdesine veya hız kodu olarak dakikada devir sayısında tanımlanabilir.

Genel olarak, son işlemci tarafından etkilenen deyimler son işlemci deyimleri ve işleme özellikleri ile ilgili deyimlerdir. APT-AC NC işlemci tarafından kabul edilebilen kelimeler son işlemci tarafından kabul edilmeyebilir veya işlenmeyebilir. Ayrıca, farklı son işlemciler tarafından işlendiği zaman (6. Bölümün 6.4'ünde açıklanan FEDRAD ve RAPID deyimleri tipik örneklerdir) bir kelime farklı anlamlara sahip olabilir. Onun için uygulanabilir son işlemci hakkında iyi bir bilgiye sahip olmalıdır; aksi takdirde doğru bir NC çıktısını elde etmede zorluklar olabilir.

hücre için tezgah işlemini tanımlayan bir APT programıdır. Bu örnek LCONIC deyiminin nasıl kullanılacağını ve özel bir profil için kesici yolunu tanımlamak maksadı ile bir döngüyü gösterir. İkinci örnek, basitleştirilmiş normal çan eğrisi gibi şekilli bir parçadır. Bu parçanın işleme operasyonu tipik bir üç boyutlu durumdur. Üçüncü örnek, Y-Z düzleminde bir kesici yolunun, X-Y düzleminde tanımlanan benzeri bir yolun dönüşümü aracılığı ile tanımlanabileceğini gösterir. Son olarak, bir tornalama işleminin programlanması 4. örnekte gösterilmektedir.

Her örnek program, bir işleme operasyonunun tanımlanmasının birçok potansiyel yaklaşımlardan birini temsil eder. Okuyucuların, bu problemleri çözmek için kendi yollarını kullanmalarını tavsiye edilir.

Örnek 1

Şekil 4-25'te gösterilen Wankel motor hücresinin profili LCONIC veya listelenmiş silindireler olarak tanımlanabilen birkaç kısma bölünebilir. LCONIC deyimini ile programlamada ilgili muhtemel problemleri ve kullanılan teknikleri açıklamak için, profilin çeşitli kısımlarını tanımlamayı kullanmaya karar verdik. Profil, X ve Y eksenlerine göre simetrik olduğu için, sadece kesici yolunun bir yarısının tanımlanmasına gerek vardır; diğer yarısı bir COPY deyimini kullanarak kopyalanabilir. Profilde 10, 20,....., 190 derece polar açılara (Şekil 8-3 [a]) karşılık gelen 19 nokta seçilir. (eğer daha yüksek hassasiyet gerekirse, daha fazla noktaya gerek vardır). Bunların koordinatları hesaplanır. P(i) hariç, P(i) (i=2n+1, n=0,1,2,.....,9) noktalarında profilin eğimleri de hesaplanır. Profilin yarısı sekiz kısma bölünür: E(1), E(2), E(3), E(4), E(6), E(7), E(8), E(9). E(4) hariç bütün kısımlar aşağıdaki deyimle tanımlanır.

LCONIC/3PT2SL

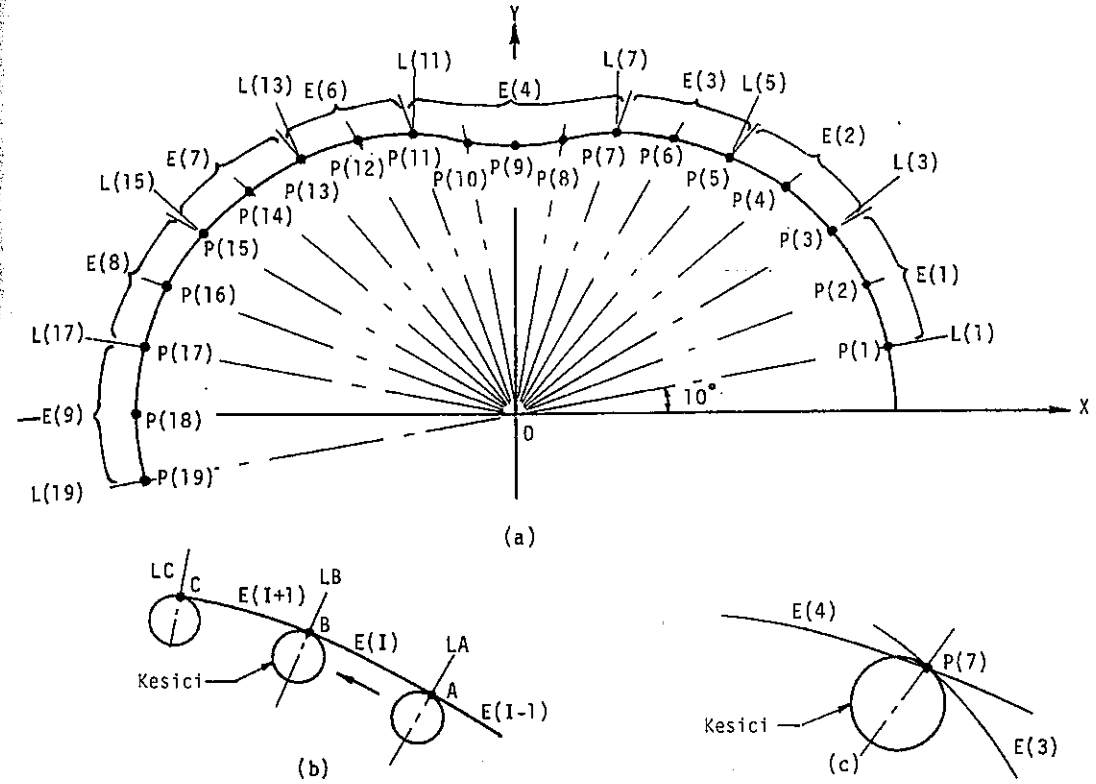
E(4) şu deyimle tanımlanır:

LCONIC/5PT,P(7),P(8),P(9),P(10),P(11)

Eğer 3PT2SL seçeneği seçilirse * P(7) noktasından P(9) noktasına olan kesici yolunun tanımlanmasında problemlere sebep olan bu kısmın özel profili (yani eğilme noktasının varlığı) nedeni ile profilin bu kısmını tanımlamak için, 3PT2SL yerine 5PT seçeneğini kullanılır. Onun için, iki uç eğimleri eğrileri tanımlamak için kullanıldığından dolayı E(4)-E(3) ve E(4)-E(6) kısımları hariç profilin herhangi iki kısmı bağlama noktalarında teorik olarak birbirine teğettir. Fakat, bağlama noktasında birbirine teğet olarak tanımlanan LCONIC'e rağmen, kesiciyi bir LCONIC, E(I), boyunca bir diğere, E(I+1) hareket ettiren aşağıdaki deyim için daima bir hata mesajı türetilmektedir.

* Eğer 3PT2SL seçeneği seçilirse, profil kısmını, P(7)-P(8)-P(9) tanımlamak için oluşturulan eşitlik;
 $0.014585131X^2 - 0.382364792XY + 2.07500844Y^2 +$
 $19.1182396x - 207.317561y + 5178.35689 - 0$ dir.

Bu P(8) ve P(9) noktaları arasında yerleştirilen merkez noktalı hiperbolik bir fonksiyondur. Bundan dolayı, bu iki nokta arasındaki profili uygun şekilde tanımlamaz.



Şekil 8.3 Şekil 4-25'te gösterilen Wankel motor hücresinin işlenmesi. (a) Hücrenin yarı profili sekiz kısma bölünür; LCONIC/5PT deyimini ile tanımlanan E(4) hariç hepsi, LCONIC/3PT2SL deyimleriyle LCONIC'lerle tanımlanır. L(1)-L(19) doğruları sırasıyla P(1)P(19) doğrularında profilin normalidirler. (b) Bir normal iki LCONIC'in birleşme noktasında tanımlanır. Bu çevresel hareket deyiminde kontrol yüzeyi olarak görev yapar. (c) P(7) noktasındaki eğimin süreksiz olmasından dolayı, kesici P(7) noktasından P(11) noktasına hareket ettiği zaman kesici, çevresel hareket kademesinin başlangıcında toleransın dışında olabilir. E(3), LCONIC/3PT2SL deyimini ile ve E(4) LCONIC/5PT deyimini ile tanımlanır.

TL..., GO...,E(I),TANTO,E(I+1)

Bu hata, bağlam noktasındaki iki LCONIC eğrisinin farklı yarıçaplarının varlığından dolayı ortaya çıkar. Bunun çözümü, her iki LCONIC'e de normal olan bir doğru tanımlamak (Şekil 8-3 [b]) ve kesiciyi konumlandırmak için onu bir kontrol yüzeyi olarak kullanmaktır. Böylece A-B-C kesici hareketi aşağıdaki gibi tanımlanabilir.

TLLFT, GOFWD/E(I),ON,LB
 GOFWD/E(I+1),ON,LC

Şekil 4-25'te ve Şekil 8-3 (a)'da gösterilen parça profilini işlemek için APT programı aşağıdaki gibi görülebilir:

```

PARTNO EXAMPLE 1
MACHIN/GN5CC, 9, OPTION, 2, 0 $$ GN5CC IS THE POSTPROCESSOR NAME
UNITS/MM
CLPRNT
CUTTER/0.0001
FEDRAT/2.0
CLPRNT
RESERV/X, 19, Y, 19, S, 19, SS, 19, E, 9, L, 19, P, 19
DO/A2, I= 1, 19 $$ CALCULATION AND DEFINITION OF POINTS P(I) AND LINES L(I)
  A=10*I
  B=3*A
  X(I)=60*COSF (A)+10*COSF(B)
  Y(I)=60*SINF (A)+10*SINF(B)
  P(I)=POINT/X(I), Y(I)
  IF ((I/2-INTGF (I/2)) 'EQ'0), JUMPTO/A2
  S(I)=- (COSF (A)+(COSF (B)) / 2) / (SINF (A)+(SINF (B)) / 2) $$ THE SLOPE
  $$ OF THE PROFILE

  IF (I'EQ'9), JUMPTO/A1
  SS(I)= -1/S(I)
  L(I)=LINE/P(I), SLOPE, SS(I)
  JUMPTO/A2
A1)  L(I)=LINE/YAXIS
A2)  CONTIN
DO/B1, J=1, 9 $$ DEFINITION OF PROFILE SECTIONS E(I)
  JJ=2*J-1
  IF (J'EQ'4), JUMPTO/B1
  IF (J'EQ'5), JUMPTO/B1
  E (J) = LCONIC/3PT2SL, P (JJ), S(JJ), P(JJ+2), S(JJ+2), P(JJ+1)
B1)  CONTIN
E(4) = LCONIC/5PT, P(7, THRU, 11)
SPINDL/700, CLW
FROM / 0, 0, 0
INDIRV / 1, 0.1, 0
GO / ON, L(1), (PLANE / 0, 0, 1, 0), E(1)
INDIRV/-0.1, 1, 0
INDEX / 7
  DO / C1, K=1, 3
  KK = 2*K+1
C1)  TLLFT, GOFWD/E (K), ON, L (KK)
  GOFWD/E(4), ON, L (11)
  DO / D1, K=6, 9
  K=2*K+1
D1)  TLLFT, GOFWD/E(K), ON, L (KK)
INDEX/7, NOMORE
COPY/7, XYROT, 180, 1

```

```

RAPID
GOTO/0, 0, 0
SPINDL/OFF
END
FINI

```

Kesici yolunun doğrulamasını basitleştirmek için bu programda çok küçük çaplı, 0.0001 mm, varsayılan bir kesici kullanılmaktadır. Gerçek işleme operasyonu için daha büyük çaplı bir kesici, örneğin 12,7 mm, kullanılmalıdır. Bu durumda, kesici P(7) noktasından P(11) noktasına hareket ettiği zaman bir problem ortaya çıkabilir. İki LCONIC'in eğimleri, E(3) ve E(4), P(7) noktasında aynı olmadığı için kesici tolerans dışında olabilir (Şekil 8-3 [c]). Böylece kesiciyi doğru olarak konumlamak için bir başlatma hareket deyimini olan

```
GO/L(7),PL1,E(4)
```

aşağıdaki

```
GOFWD/E(3),ON,L(7)
```

ve

```
GOFWD/E(4),ON,L(11)
```

hareket deyimlerinin arasına konmalıdır:

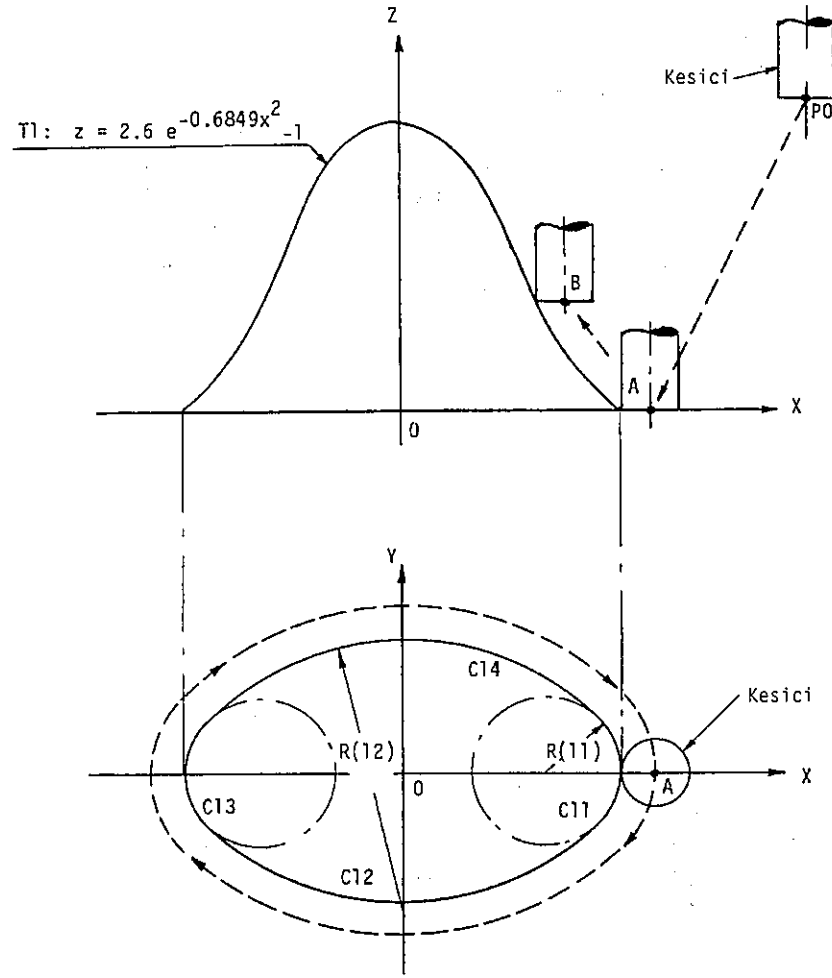
Başlatma hareket deyiminin hareket yönünün de kontrol edilmesine itina gösterilmelidir. Eğer bir sonraki deyimden farklı ise, bir sonraki deyim için hareket yönü doğru olarak tanımlanmalıdır. Gerçekte, profil ve eğim sürekliliği kolayca muhafaza edilsin diye profili tek bir listelenmiş silindir olarak tanımlayabiliriz.

Örnek 2

Şekil 8-4, z ekseninden geçen dikey kesitinde normal çan eğrisi profilli bir parçayı göstermektedir. Problemi basitleştirmek için, her seviyedeki yatay kesit, dört çember yayından oluşan bir yarı elips olan şekilde gösterilen kesite benzer olduğu kabul edilmiştir. Ayrıca, 0.5 inç çaplı düz uçlu bir kesici kullanılmaktadır. Kesicinin başlama noktası P(5,0,2)'dir.

Kesici yolu aşağıdaki gibi tanımlanmaktadır:

1. Kesici, çevresel harekete başlamaya hazır A konumunda konumlandırılır. Bu başlatma hareketi için tahrik, parça ve denetim yüzeyleri sırası ile X-Y düzlemi, T1 listelenmiş silindiri ve yatay kesitin parçası olan C11 çemberidir.
2. Kesici, yatay düzlemde, tahrik ve kontrol yüzeyleri olarak C11, C12, C13 ve C14'a kullanarak hareket etmek için talimat alır ve bu hareket sonunda A noktasına geri gider.



Şekil 8.4 Üç boyutlu bir parça, X-Z düzlemindeki kesiti, T1 listelenmiş bir silindir olarak tanımlanan bir normal çan eğrisinin şekline sahiptir. Onun herhangi bir yatay düzlemdki kesiti dört çember yayından oluşan yan elipse benzerdir. R(11)-0,3937; R(12)-1.3780.

3. Kesici, B konumuna hareket etmek için talimat alır. Bu harekette tahrik, parça ve kontrol yüzeyleri sırası ile X-Y düzlemi, T1 ve bir sonraki yatay kesitin sürekli çemberlerinden biri olan C21 (Şekilde gösterilmedi)'dir.

Bu işlem kesici tepe noktasına ulaşıncaya kadar tekrarlanır. Bu işlem için program aşağıda verilmektedir.

```

PARTNO EXAMPLE 2
MACHIN/GN5CC, 9, OPTION, 2, 0
CLPRNT
NOPOST
RESERV/X, 16, Z, 16, R, 162, C, 164
CUTTER/0.5
FEDRAT/2.0
$$ UNITS/INCHES
DO / A1, I=1, 16
    Z(I)=(I-1)*0.1
A1)  X(I)=SQRT((-LOGF((Z(I)+1) / 2.6 / 0.6849)
T1=TABCYL/NOY, SPLINE, Z (1), X(1), Z(2), X(2), Z(3), X(3), Z(4), X(4), Z(5), X(5), $
      Z(6), X(6), Z(7), X(7), Z(8), X(8), Z(9), X(9), Z(10), X(10), Z(11), $
      X(11), Z(12), X(12), Z(13), X(13), Z(14), X(14), Z(15), X(15), Z(16), $
      X(16), 1.6, 0
PL1=PLANE / 0, 1, 0, 0    $$ PLANE Y=0
PL2=PLANE / 1, 0, 0, 0    $$ PLANE X=0
DO/A2, J=1, 16
    J1=10*J+1
    J2=10*J+2
    J3=10*J+3
    J4=10*J+4
    R(J1)=(0.3937/X(1))*X(J)
    R(J2)=(1.3780/x(1))*X(J)
    C(J1)=CIRCLE/(X(J)-R(J1)), 0, R (J1)
    C(J3)=CIRCLE/(R(J1)-X(J)), 0, R (J1)
    C(J2)=CIRCLE/YLARGE, IN, C(J1), IN, C(J3), RADIUS, R(J2)
A2)  C(J4)=CIRCLE/YSMALL, IN, C(J1), IN, C(J3), RADIUS, R(J2)
SPINDL/600, CLW
FROM/5, 0, 2
INDIRV / -1, 0, -0.3
GO / ON, PL1, T1, TO, C(11)
DO/C1, K=1, 16
    K1=10*K+1
    K2=10*K+2
    K3=10*K+3
    K4=10*K+4
    KK1=10*(K+1)+1
    INDIRV/0, -1, 0
    AUTOPS
    ARCSLP/ON;    TLLFT, GOFWD / C(K1), TANTO, C(K2)
                  GOFWD / C(K2), TANTO, C(K3)
                  GOFWD / C(K3), TANTO, C(K4)
                  GOFWD / C(K4), TANTO, C(K1)
                  GOFWD / C(K1), ON, PL1
GO/ON, PL1, T1, C(K1)
IF (K'EQ'16), JUMPTO/C1
TLON, GOUN/PL1, C(KK1)

```

```

C1) CONTIN
TLON, GOUP/PL1, ON, PL2
RAPID
GODLTA/1.0
SPINDL/OFF
END
FINI

```

Örnek 3

Şekil 8-5 (a)'da gösterilen parça, koordinat dönüşüm yöntemi aracılığı ile bir kesici yolunun tanımlanmasını göstermek için tasarlandı. İşlenmemiş parça $7 \times 3.5 \times 2.5$ inç ölçülerindeki bir dikdörtgenler prizmasıdır. X-Z düzlemine paralel düzlemlerde dairesel biçimde kesiti olan düzenli olarak azaltılan fazla talaş payı çapı 0.5 inç olan bir yuvarlak uçlu parmak freze ile bir freze tezgahında işlenecektir. X-Z düzlemine paralel dairesel kesitlerin merkezleri 90 dereceye eşit olan dairesel yay açılı L1 doğrusu üstündedir. Kesicinin başlama noktası P(0,0,1.0)'da seçilir ve gerekli kesici yolu $A_1 - A_2 - A_3 - A_4 \dots$ 'dür. ($A_1 - A_2$ ve $A_3 - A_4$ yolları, X-Z düzlemine paralel bir düzlemde daireseldir). Dairesel kesitler X-Z düzleminde olduğu için, CYLNDR veya TABCYL deyimlerini bu kesitleri tanımlamak için kullanmak oldukça hantal olur. Bu nedenle X-Y düzleminde geometrik öğeler ve kesici yolu tanımlandı ve sonra onların gerçek konumlarına dönüşümleri yapıldı (Şekil 8-5 [b]). Program aşağıdaki gibidir:

```

PARTNO EXAMPLE 3
MACHIN / GN5CC, 9, OPTION, 2, 0
NOPOST
RESERV / R, 11, C, 11, L, 111
CLPRNT
$$ UNITS / INCHES
CUTTER / 0.25 $$ THE BALL-END MILL IS IMITATED BY A FLAT-END MILL.
FEDRAT / 2.0
L1=LINE / (0, 0, 0), (1.0, 1.0, 0)
DO / A1, I=1, 11
    I1=I*10+1
    R(I)=(1.5+(I-1)*0.1)*SQRTF (2)
    X=R(I) / SQRTF (2)
    C(I)=CIRCLE / CENTER, X, X, 0, RADIUS, R(I)
A1) L(I)=LINE / (POINT 7 CENTER, C(I), (POINT / (2*X), 0, 0) $$ SEE FIG. 8-5B
PL3=PLANE / 0, 0, 1
M1=MATRIX/TRANSL, 0, 0, -.125, TIMES, YZROT, 90
SPINDL/900, CLW
FROM/0, 0, 1, 0
DNTCUT
GODLTA/1, 1, 0
INDIRV / -1, -1, -1
TRACUT / M1

```

```

GO/ON, L1, PL3, C(1)
CUT
INDIRV / 1, -1, 0
ARCSLP/ON
TLLFT, GOFWD / C(1), ON, L(11)
TRACUT/LAST, (MATRIX/TRANSL, 0, 0.35, 0)
INDRIV / 1, 0, 0
GO/ON, L(21), PL3, C(3)
TLRGT, GORGT 7 C(2), ON, L1
TRACUT/LAST, NOMORE
TRACUT/NOMORE
RAPID
GOTO/0, 0, 1
SPINDL/OFF
END
FINI

```

Bu programda sadece kesici yolunun bir kısmının, P0-A1-A2-A3-A4-P0 tanımlandığına dikkat ediniz. Okuyucuya istenen profili işlemek için eksiksiz bir program yazması tavsiye edilir.

Örnek 4

Şekil 8-6'da gösterilen mil sadece iki eksenli, X ve Z olan bir tornada işlenmektedir. Kesici yolu X-Y düzleminde tanımlanır ve sadece X ile Z kodlarından ibaret olan bir SD programa son işlemci tarafından değişime tabi tutulur. Bitirme işlemi işlemci tarafından değişime tabi tutulur. Bitirme işlemini tanımlamak için program aşağıdaki gibi hazırlanabilir:

```

PARTNO EXAMPLE 4
MACHIN/LATH1 ..... $$ THE LATHE POSTPROCESSOR
CUTTER / 0.02 $$ THE CUTTER NOSE RADIUS IS 0.01.
$$ UNITS/INCHES
FEDRAT/IPR, 0.005
COOLNT/ON
SPINDL/750, CCLW $$ THE SPINDLE ROTATION DIRECTION DEPENDS ON THE
$$ STRUCTURE OF THE MACHINE TOOL.
LOADTL/1, SETOOL, 0, 0, 0
.....
..... (L1'den L11, C1 ve P0 başlama noktasını tanımlayan
..... geometrik tanımlama deyimleri)
.....
$$ THE FOLLOWING STATEMENTS DEFINES THE FINISHING CUT.
FROM/P0
RAPID
GO/L3, (PLANE / 0, 0, 1, 0), L1

```

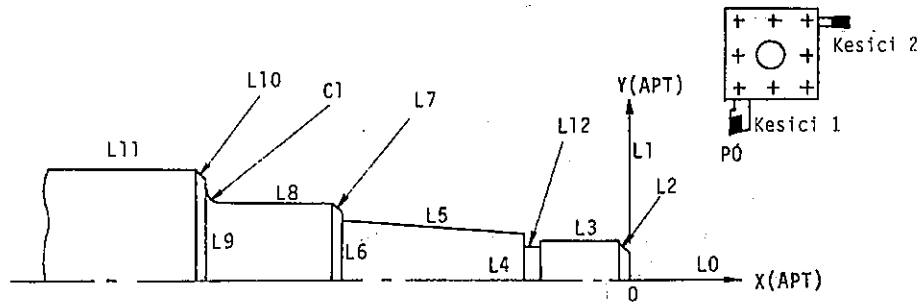

GO/ON, L5, (PLANE/0, 0, 1, 0), L4
 FEDRAT / 0.002
 TLLFT, GOLFT / L4, L12
 RAPID
 GODLTA / 0, 2.0, 0
 RAPID
 GOTO / P0
 SPINDL/OFF
 COOLNT/OFF
 END
 FINI

8.5 IBM APT-AC NÜMERİK KONTROL İŞLEMCİSİNE GİRİŞ

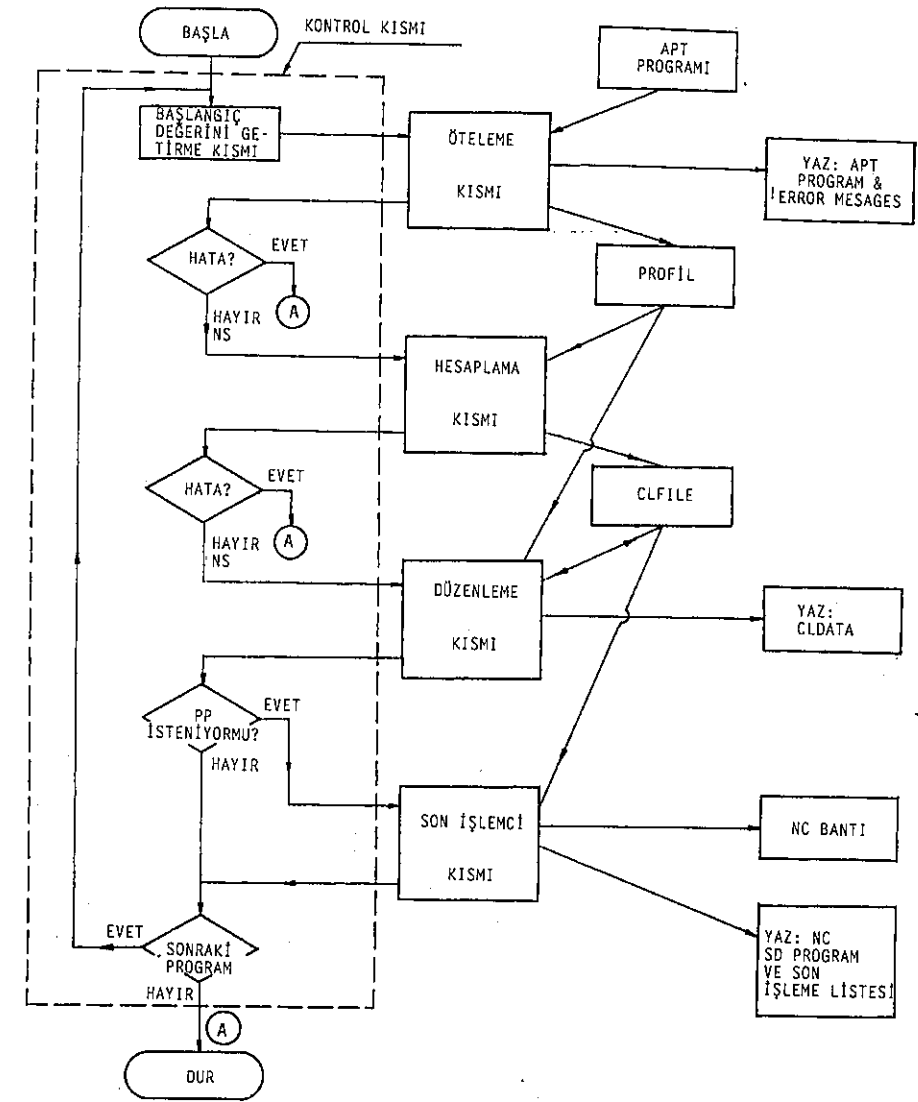
IBM APT-AC NC işlemcisi, kullanıcının yazdığı APT programını girdi olarak kabul eden, işleyen ve işlenmiş sonucu standart bir formata veya CLDATA diye bilinen ortak arabirim koduna dönüştüren bir programdır. NC işlemci makinarya dayalı son işlemcinin geliştirilmesine de destek sağlar. Daha sonra MACHIN deyimi aracılığı ile programcının isteğine bağlı olarak CLDATA son işlemci tarafından işlenir. Nihai sonuç NC kodunda makina-okuyabilir bir programdır.

APT-AC NC işlemcisinin genel yapısı Şekil 8-7'de gösterilmekte ve aşağıdaki ana kısımlardan oluşmaktadır.

1. Denetim kısmı
2. Çeviri kısmı
3. Hesaplama kısmı
4. Düzeltme kısmı
5. Son işlemci kısmı



Şekil 8-6 Torna tezgahında işlenen bir mil.



Şekil 8-7 IBM APT-AC NC işlemcisinin genel yapısı. Not: NS-ciddi hata yok, PP-son işlem.

Bir APT programı, sırası ile çeviri, hesaplama, düzenleme ve son işlemci kısımları tarafından denetim kısmının denetimi altında işleme tabi tutulur. Bir iş, işlemci kısmına denetim kısmı tarafından atanır. Bu tamamlandıktan sonra, denetim, denetim kısmına geri döner, daha sonra bu işleme işi bir sonraki kısma atanır.

Çeviri kısmının temel fonksiyonu APT'de tanımlanan parça geometrik öğelerini, APT deyimlerine diğer tipleri ile birlikte kabul edilen (standart) biçimlere çevirip, daha sonra PROFIL diye adlandırılan bir ara dosya olarak saklamaktır. Bir REFSYS deyimini ile tanımlanan bir dönüşüme rağmen geometrik öğelerin bu safhada dönüşüme tabi tutulmadığına dikkat edilmelidir. REFSYS deyimlerinde tanımlanan matrisler saklanır ve bir sonraki işlem (hesaplama) safhasına aktarılır.

Hesaplama kısmı, PROFIL dosyasını girdi olarak kullanır ve kesici uç merkezinin birbirini takip eden konumlarını hesaplar. Son işlemci, kelimeleri ve komutları ile birlikte bu konumlar, CLFILE diye adlandırılan yeni bir dosya oluşturur. Bu kısımda, REFSYS deyimleri ile tanımlanan dönüşümlerin icra edildiğine; fakat, TRACUT ve COPY deyimleri ile tanımlananlar düzenleme kısmı çağrılana kadar geciktirildiğine dikkat ediniz. Daha sonra eğer dönüşümler tanımlanırsa, düzenleme kısmı kesici uç merkez konumlarını dönüşüme tabi tutar ve sonuç yeniden düzenlenmiş bir CLFILE'dir. CLFILE'in yazılı biçimi CLDATA diye adlandırılır ve parçayı işlemek için gerekli kesici konumları ve APT programında tanımlanan son işlemci komutları ile ilgili verilerden ibarettir.

Son işlemci kısmı, kullanıcı tarafından oluşturulan bir veya bir kaç son işlemciden ibarettir. O, CLDATA'yı işlemek için APT programındaki MACHIN deyimine göre gerekli son işlemciyi seçer. Son işlemcinin çıktısı, bilgisayar sisteminin hafıza biriminde saklanabilen ve tanımlanan çıktı cihazına bağlı olarak yazılmış biçimde (liste dosyası) veya NC şeridi olarak çıktısı verilebilen gerekli makina okuyabilir NC programıdır.

Her işleme safhası esnasında, APT işlemcisinin ilgili kısmı denetim kısmı ile iletişimini sürdürür; eğer programda ciddi bir hata tespit edilirse denetim erken olarak denetim kısmına geri gönderir. Sonra işleme durdurulur ve hataların tipi ve yapısına bağlı olarak hata mesajı verilir. Eğer hatalar işlemeyi durduracak kadar ciddi değilse, sadece hata mesajları verilir ve işlem devam eder.

3 ile 7. Bölümler arasında açıklanan fonksiyon ve kapasitelere ek olarak IBM APT-AC NC işlemcisi kullanıcıya aşağıdaki kabiliyetleri de sağlar:

1. Parça program dosyalarını geometrik tanımlama dosyalarını ve işlenebilirlik dosyalarını düzenlemek ve NC verileri tabanı olarak görev yapan ve girdi olarak kullanılabilen veya bir APT programı tarafından çağrılabilen APT kütüphanesinde, bu dosyaları saklamak.
2. Kütüphane muhtevasına silme, ilave ve yer değiştirme yapılabilir. (PROC) işlemi gibi mantık talimatlarının bir serisini tanımlamak; bunu kütüphane de saklamak ve sonra tekrarlamalı olarak icra etmek.
3. Kesici yönlendirmesinde (çok eksenli programlamada) değişiklikleri gerektiren işleme yöntemi için kesici yolunu tanımlamak.

4. Genişletilmiş PRINT deyimleri aracılığı ile, örneğin, tanımlamaları, parça program listelerini ve sembol referans listelerini yazmak.

1985'te IBM şirketi, aşağıdaki ek özellikleri içine alan APT-AC NC işlemcinin yeni bir yayını (uyarlama 1, yayın 4, düzeltme seviyesi 0, Program No. 5740-M53) neşretti⁷.

1. Önceki yayın tarafından kabul edilen geometrik öğeleri tanımlamak için yeni yollar.
2. İşleme amaçları için noktaları, doğru parçalarını, çember yaylarını ve çemberleri bir tek bileşik eğriye birleştirme kabiliyeti.
3. İki boyutlu spline, rasyonel B-spline, üç boyutlu sınırlı doğru veya herhangi bir yönde çember yayı gibi yeni geometrik öğelerin tanımlanması.
4. İşlenmemiş parça ve parça porfillerini tanımlama ve sonra parçayı torna-lamak için kaba ve ince işleme kesici yollarını türetme kabiliyeti.

APT-AC NC işlemci programları, hem FORTRAN'da ve hem de Assembler dillerinde yazılmıştır. Bunlar, en az 2.5 megabyte zahiri saklama kapasiteli herhangi bir IBM Sistemi/370 veya VM/CMS veya OS/MVS işletim sistemi çevresi altında çalıştırılabilirler. Bilgisayar sistemi, NC işlemcisi ve bunun çalışma ihtiyaçları için tahsis edilmiş ek saklama birimine sahip olmalıdır. Normal olarak, gerekli ek saklama kapasitesi, yaklaşık bir-üç arası Model 3350 silindir gerektiren NC işlemcinin kendisi (yaklaşık 46 model/3350 silindir veya karşılığı), iş ve kütüphane dosyası (iki-beş arası Model 3350 silindir veya karşılığı), girdi (APT program) ve çıktı (NC program ve çıktısı (listesi)) dosyalarına tahsis edilen alanın toplamıdır. Toplam gerekli yardımcı saklama biriminin kapasitesi, yaklaşık 59-54 arası IBM Model 3350 silindiridir.

8.6 BİR APT PROGRAMININ ÇALIŞTIRILMASI İŞLEM SIRASI

Yazılı bir APT programı NC işlemci tarafından işlenmesi için bilgisayar sistemine girilmelidir. Bu girdi dosyasının formatı aşağıdaki gibi olmalıdır:

1. Her satır (veya kart) 80'e kadar karakter içerir.
2. Bir APT deyimini sadece 1072 sütunları arasında yazılabilir veya delinebilir.
3. 73-80 arası sütunların içeriği NC işlemci tarafından ihmal edilir. Bu sütunlar, tanıma ve sıraya koyma amacı için olan alfasayısal karakterleri tanımlamak maksadı ile kullanılabilir.

Bir satır (veya kart) bir noktalı virgül veya noktalı virgüllerle bölünmüş birden fazla

APT deyimi içerebilir.

Bir APT programına bir isim verilmeli ve doğrudan erişimli hafıza biriminde saklanmalıdır. Böylece o, çağrılabilir, düzeltilebilir ve işlenebilir.

Bir APT programının IBM VM/CMS işletim sisteminde çalıştırılması için detaylandırılmış işlemler aşağıdaki gibidir:

1. IBM APT-AC NC işlemci programı, FORTRAN-H genişletilmiş programı ve diğer ilgili kütüphane bilgilerinin olduğu diskleri birbirine bağlayınız. Sonra ulaşılabilirlik amacıyla bu bağlı diskler için adresleri tanımlayınız.
2. NC işlemci operasyonu için gerekli olan NC işlemci teşhis tablosunu hazırlayınız ve onu bir dosya olarak saklayınız. Örneğin;

LP LP A4

sırasıyla teşhis tablo dosyasının ismi, tipi ve modu için sembollerdir.

3. Gerekli son işlemci programlarını (veya modüllerini) bir kullanıcının A hesap diskinde ise, son işlemcileri saklayan diski A kullanıcı diskinde saklama yerine yukarıdaki 1. yöntemle göre kullanıcının numarasına bağlamak daha iyidir.
4. İşlenecek bir APT programı yazınız ve saklayınız. Dosya adı, tipi ve saklanan dosyanın modu, örneğin, sırası ile olabilir.

TEST PART A4

5. APT Programının işlemek için kullanılan aşağıdaki iş icra programını yazınız:

&CONTROL ALL

FI FT05F001 DISK &1 &2 A4 (RECFM F BLOCK 80)

FI ERRFILE DUMMY

FI FT06F001 DISK LT1 LT1 A1

FI FT01F001 DISK F1 F1 (RECFM F BLKSIZE 3228)

FI FT02F001 DISK F2 F2 (RECFM F BLKSIZE 3228)

FI DPOOL DISK SP SP (XTENT 50 RECFM F BLKSIZE 4096)

FI LIBDPOOL DISK LP LP A4 (RECFM F BLKSIZE 4096 DISP MOD)

FI SYSPUNCH DUMMY

FI POSPUNCH DISK LT3 LT3 A1

FI SOURCE DISK PARTPDS PARTPDS A4 (RECFM F BLOCK 80 DISP MOD)

FI 7 TERM

LOADMOD DKWOR6

START * LINELNTH 133 LINECNT 76 POOLSIZE 10 5

Bu iş icra programı, aşağıdaki düzeltmelerle IBM APT AC-SD işlemcisindeki ⁸ ACGOR6 dosyasından çıkartılmıştır:

- a. İşlem listeleme dosyası A diskinde LT1 LT1 dosyası olarak saklanır.
- b. A diskinde POSPUNCH son işlemci çıktı dosyası (yani SD programı) LT3 LT3 dosyası olarak saklanır.

c. SYSPUNCH delinmiş çıktı ihmal edilmektedir.

FT05F001 dosyası SD işlemciye girdi dosyasıdır (yani APT programı). "&1" ve "&2" iki sembolleri APT programının dosya adı ve dosya modunu temsil eder. Bu düzeltmeler hata ayıklama ve programı düzeltmeyi kolaylaştırmak için yapılmıştır. Eğer doğru bir APT programı elde edildikten sonra delinmiş çıktıya gerek duyuluyorsa, NC işlemcideki ACGOR6 dosyası kullanılabilir.

Son deyimdeki, ilk iki parametre, START, çıktı dosyasının formatını ayarlamak için değiştirilebilir. LINELNTH seçeneği için olan parametre yazılı çıktının satır uzunluğunu tespit eder. Bu ya 80 veya 133 olabilir (her satırdaki yazı karakterleri). Bazı çıktılar, her satır için 80 karakter modunda yazılamayacağı için normal olarak 133 parametresi kullanılır. LINECNT seçeneğinin parametresi çıktının her sayfasındaki yazılacak en çok satır sayısını tanımlar. Bu 2 ile 132 arasında herhangi bir tamsayı olabilir. Varsayılan değer 55'tir.

Okuyucu yukarıda listesi verilen diğer deyimlerin anlamları için uygun IBM el kitaplarına ^{7,8} başvurmalıdır.

Bu iş icra deyimleri, iş icra program dosyası olarak saklanabilir. Dosya adı, diğer iş icra dosyaları isimleri ile veya APT sisteminde kullanılan dosya isimleri ile aynı olmamak şartı ile rastgele olabilir. Dosya tipi, EXEC (iş icra program dosyası olduğunu gösteren) olmalıdır. Dosya modu A'yı takip eden 0'dan başka herhangi sayısal karakterden oluşan bir sayıyla isimlendirilebilir. Böylece; örneğin, yukarıdaki programı aşağıdaki dosya adı, tipi ve modu altında saklanabilir.

PROAPT EXEC A1

6. APT programını işlemek için şunu bilgisayarda yazınız:

PROAPT TEST PART

İş icra programındaki iki sembol otomatik olarak TEST ve PART tarafından değiştirilir. Böylece APT programı işlenir. Hataların boyutuna bağlı olarak LT1 LT1 A1 ve LT3 LT3 A1 dosyalarından biri veya her ikisi türetilecektir. Eğer hata yoksa veya sadece bazı küçük hatalar varsa bu iki dosya, LT1 LT1 A1 liste dosyasında yazılan hata mesajlı ve mesajsız türetilecektir. Eğer hatalar ciddi ise LT3 LT3 A1 SD çıktısı türetilmez. Genellikle ilk işleme anında dosya listelenirken birkaç hata türetilir.

7. Programı hata mesajlarına göre düzeltiniz ve yukarıdaki 6. safhaya göre onu yeniden işleme tabi tutunuz. Bu işlemi APT programındaki tüm hatalar düzeltilinceye kadar tekrarlayınız.


```

.....
.....
CUTTER / 0.25
R=0.125
R1=1.9223
C1=CIRCLE / CENTER, 1.5625, 0, RADIUS, R
C2=CIRCLE / CENTER, 0, 1.5625, RADIUS, R
C3=CIRCLE / CENTER, -1.5625, 0, RADIUS, R
C11=CIRCLE / YLARGE, OUT, C1, OUT, C2, RADIUS, R1
C22=CIRCLE / YLARGE, OUT, C2, OUT, C3, RADIUS, R1
FROM / (P0=POINT / 0, 0, 0)
GO / C11
AUTOPS
TLLFT, GOLFT/C11, TANTO, C2
A1) GOFWD/C2, TANTO, C22
.....
.....

```

A1 etiketli deyim için iki hata mesajı verilir:

```

ERROR 2210: "WARNING ..... CUTTER IS ON THE WRONG SIDE OF DRIVE SURFACE,
THE PROCESSING CONTINUES" *
ERROR 2619: "UNABLE TO POSITION THE CUTTER INSIDE OF CIRCLE; CHECK
CUTTER AND CIRCLE RADII" **

```

Anlamları: * "İkaz! Kesici, tahrik yüzeyinin yanlış tarafında. İşlem devam ediyor".
** "Kesicinin çember içine konumlanması mümkün değil. Kesici ve çember yarıçaplarını kontrol ediniz".

Kesici çapı, C2 çemberi ile aynı olduğu için bu iki hata ortaya çıkar. 6. Bölümde belirtildiği gibi, kesici yolu her zaman gerçek profilin doğrusal enterpolasyonudur ve P2'deki profilden belli sapmalar kaçınılmazdır (Şekil 8-9 [b]). Bu programdaki tolerans +0.0005 inç olan önceden varsayılan değerdir. Onun için kesici diğer kenarda C2 çemberi ile muhakkak girişime uğrar ve böylece kesici konumu P2'de tam olarak teorik konumda olmadıkça kesici asla C2 çemberi içinde konumlandırılmaz.

Çözüm, kesici çapını küçültmektir. Sonuç aşağıdaki tablodan görülebilir:

KESİCİ ÇAPI (inç)	TÜRETİLEN HATA(LAR)
0.2500	2210 ve 2619
0.2490	sadece 2619
0.2480	hata yok

Yukarıda verilen iki örnek, hataya çözümün hatanın tespit edildiği deyimde olmayacağına açıkça göstermektedir. Bütün muhtemel durumları burada listeleme-

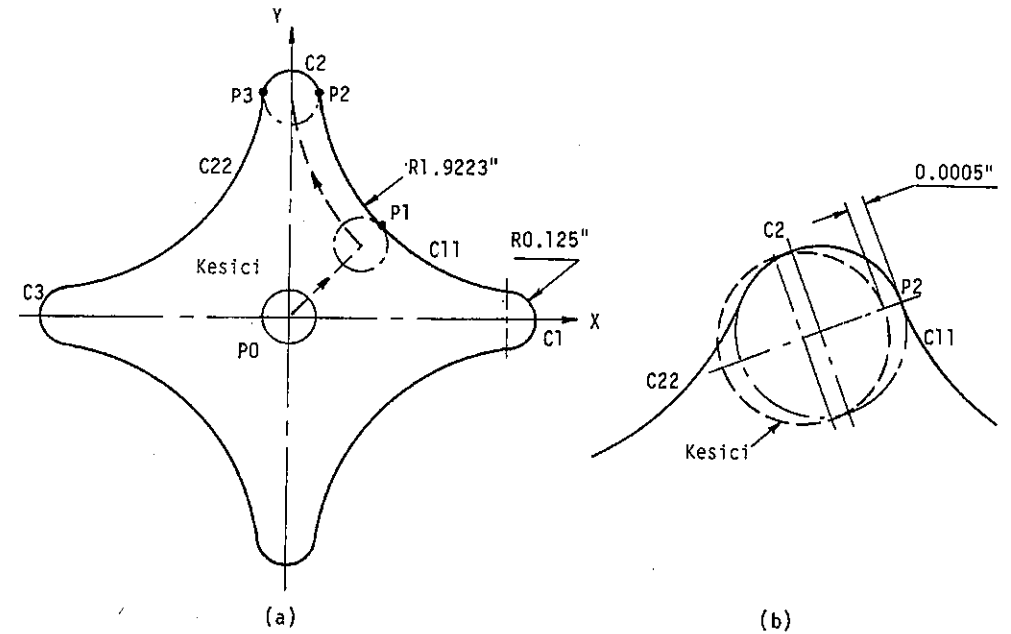
ye imkan olmamasına rağmen, bu bölümde verilen ilkelerin bir programın hata ayıklamasında faydalığını ispatlamaktadır.

8.8 İŞLEM SIRASI LİSTELEME DOSYASI VE CLDATA

Bir APT programının işlemesi bittikten sonra, istenirse APT-AC NC işlemcisi, CLDATA dosyasını ve listeleme dosyasını üretir. Bu iki dosyanın tam olarak anlaşılması, bir APT programında hata ayıklamaya yardımcı olur. Bölüm 8.7'deki Örnek 1'de tartışılan program için listeleme dosyası ve CLDATA Şekil 8-10'da listelenmiştir.

Liste dosyasının ilk parçası 1. kısmın (çeviri kısmı) çıktısıdır. Bu herbirinin bir ISN numarasının atandığı APT deyimlerinin bir listesinden oluşur. Deyim listesinin sonunda, NC işlemcisi çeviri öteleme safhası esnasında herhangi bir hatanın tespit edilip edilmediğini gösterir. Eğer bulunursa detaylı hata mesajları verilir. Hata ciddi ise, işlem ve (bağlı olarak) liste dosyası durdurulur.

Hata tespit edilmezse, normal olarak NC işlemcisinin hesaplama ve düzeltme kısımları bir çıktı vermez. CLPRNT deyimini isteği üzerine, NC işlemcinin düzeltme kısmı 3. kısımda (düzeltme kısmı) CLDATA çıktısını yazar.



Şekil 8.9 (a) Örnek 2'de tartışılan kesici yolu. (b) profilden kesicinin normal sapması.

....ÇEVİRİ SAFHASINA BAŞLA... (KISIM 1)

```

ISN 00001 PART NO TEST ERROR
ISN 00002 CLPRNT
ISN 00003 UNITS / INCHES
ISN 00004 CUTTER / 0.5
ISN 00005 FEDRAT / 2.0
ISN 00006 SPINDL / 500, CLW
ISN 00007 PL1=PLANE / 1, 0, 0, 5
ISN 00008 PL2=PLANE / 0, 1, 0, 0
ISN 00009 PL3=PLANE / 0, 1, 0, 5
ISN 00010 PL4=PLANE / 0, 0, 1, 0
ISN 00011 C1=CIRCLE / CENTER, 5, 0, 0, RADIUS, 2.0
ISN 00012 A1) FROM / 0, 0, 0
ISN 00013 A2) GO / C1
ISN 00014 A3) TLLFT, GOLFT / C1, PL1
ISN 00015 A4) GOLFT / PL1, PL3
ISN 00016 SPINDL / OFF
ISN 00017 END
ISN 00018 FINI

```

ÇEVİRİ SAFHASINDA TEŞHİSLER ALINMADI
18 N/C SOURCE RECORDS (SYSIN)

KISIM 1 GEÇEN CPU ZAMANI 0000/00.1199 DAK/SANIYE
KISIM 2 GEÇEN CPU ZAMANI 0000/00.0699 DAK/SANIYE

...KISIM 3...

```

ISN 0001 PART NO TEST ERROR
0003 UNITS / INCHES
0004 CUTTER / 0.500000000
0005 FEDRAT / 2.000000000
0006 SPINDL / 500.000000000 CLW
0012 FROM / 0.0 0.0 0.0 0.0

```

```

LABEL REC M CARD
00002 E0001
00004 E0003
00006 E0004
00008 E0005
00010 E0006
A1 00012 E0012

```

Şekil 8-10 Kısım 8.7'deki Örnekle 1'de gösterilen örnek program ve onun CLDATA'sı. (devamı var)

```

0013 GOTO / C1 2.750000000 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0014 SURFACE C1 5.000000000 0.0 0.0 0.0 1.000000000 2.000000000
0014 GOTO/ C1 2.74998912 0.04691809 0.0
2.75390084 0.14067270 0.0
2.76171747 0.23418274 0.0
2.77342542 0.32728565 0.0
2.78900435 0.41981957 0.0
2.80842716 0.51162361 0.0
2.83166009 0.60253819 0.0
2.85866274 0.69240523 0.0
2.88938818 0.78106850 0.0
.....
3.98925546 2.01075750 0.0
4.07395579 2.05114416 0.0
4.16026608 2.08796484 0.0
4.24803627 2.12115553 0.0
4.33711377 2.15065853 0.0
4.42734373 2.17642253 0.0
4.51856926 2.19840276 0.0
4.61063178 2.21656099 0.0
4.70337122 2.23086567 0.0
4.75000000 2.23686798 0.0
PL1
4.75000000 4.75000000 0.0
OFF
0015 GOTO / PL1
0016 SPINDL /
0017 END
0018 ****FINI****
.... END OF SECTION 3 ....

```

KISIM 3 GEÇEN CPU ZAMANI 0000/00.0799 DAK/SANIYE
TOPLAM PARÇA PROGRAMI CPU ZAMANI 0000/00.2699
**** APT İŞLEMİNİN SONU ****

Şekil 8-10 (devamı var)

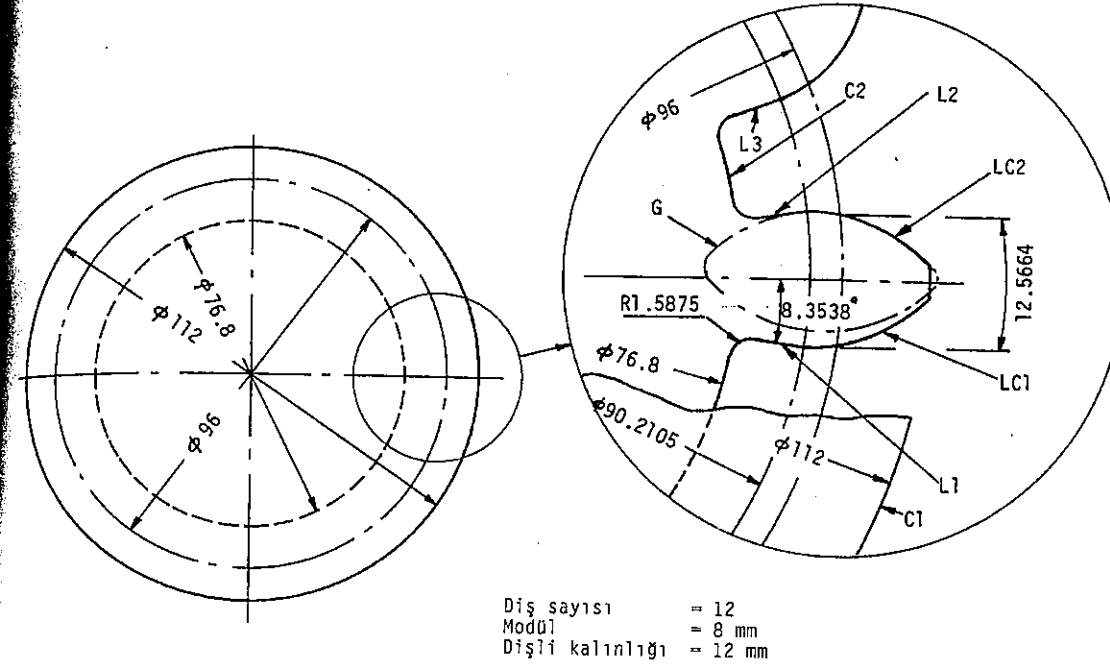
CLDATA'nın yapısı aşağıdaki gibidir:

1. CLDATA'nın içeriği, hesaplanan birbiri peşi sıra gelen kesici ekran çizgisi konumları ve APT programındaki bilgisayar operasyonları ile ilgili olanlardan başka deyimlerden ibarettir.
2. Bazı APT deyimleri için, CLDATA'da bir çıktı satırı türetilir; diğerleri için olan çıktı birden fazla satırı işgal edebilir. Bir CLDATA satırı, bağlı bulunduğu deyimi gösteren bir ISN numarası ile başlar. Bunu deyim içeriği takip eder. Programda tanımlanırsa, referans için etiket deyim (LABEL) ve deyim sıra numarası da (CARD) satır sonunda listelenir. Bir MACRO çağırma deyimi ne zaman verilirse verilsin, bir M karakteri, M sütununun altında görünecektir. CLDATA'daki kayıt numarası REC sütununda listelenir.
3. Bir hareket deyimi için, CLDATA GOTO kelimesi ile başlar ve bunu deyimde tahrik yüzeyi olarak tanımlanan geometrik öge sembolü takip eder. Noktadan-noktaya ve başlatma hareket deyimleri için, hareketin son noktası bir sonraki satırda listelenir (Şekil 8-10'daki ISN 0013'e bakınız). Çevresel hareket deyimleri için, tahrik yüzeyi GOTO kelimesinden önce kurala uygun bir biçimde verilir (Şekil 8-10'daki ISN 0014'e bakınız) ve daha sonra doğrusal enterpolasyon hesaplamasına dayalı olarak bir biri peşi sıra gelen noktaların X,Y ve Z koordinatları listelenir. Son koordinat seti bu çevresel hareketin son noktasını temsil eder.
4. COPY ve TRACUT deyimleri tanımlanırsa, bunlar da CLDATA'da listelenir.

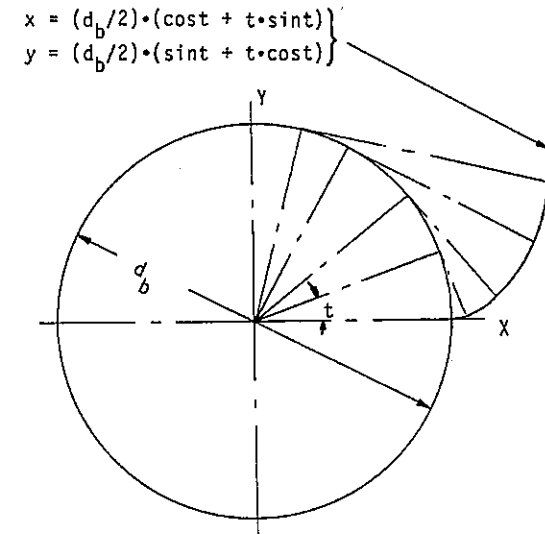
CLDATA, hatalı programın hatasını ayıklamada ve NC işlemcisi tarafından verilen hata mesajı olmayan bir programın doğrulanmasında çok faydalıdır. Bu program söz diziliminde ve mantıkta doğru olabilir ve böylece NC işlemcisi tarafından işlendikten sonra hata mesajı türetmeyebilir. Fakat kesici yolu hala yanlış olabilir, çünkü geometrik öğeler ve kesici hareketi yanlış olarak tanımlanmıştır.

PROBLEMLER

- 8.1 Problem 2.3'te açıklanan parça için bir APT programı yazınız. Başlama ve bitiş noktaları (200, 200, 50)'dir. Delik ve dış profilin bitirme işlemi için izin verilen pay 0,5 mm, tolerans 0.02 mm'dir.
- 8.2 12 dişli bir dişli çark (Şekil P8-2 [a]) üç eksenli bir NC'li freze tezgahında işlenecektir. Dişli çarkın diş profili, bir involüt eğrisi, LC1 veya LC2, involüt eğrisinin (90.2105 mm'lik çaplı) taban çemberi ile kesişmesindeki involüt



Şekil P8-2(a)



Şekil P8-2(b)

eğrisine teğet olan bir doğru, L1 veya L2 ve (1.5875 mm'lik yarıçaplı) bir köşe çemberinden oluşan üç parçadan ibarettir. Bir involüt eğri bir ipin taban çemberinden açılırken bir ipin uç noktasının konumudur. Parametrik involüt fonksiyonları ile tanımlanır:

$$x = (d_b/2) \cdot (\cos t + t \cdot \sin t)$$

$$y = (d_b/2) \cdot (\sin t - t \cdot \cos t)$$

buradaki d_b taban çemberinin çapıdır. Invölüt eğrisinin sadece taban çemberinin dışında tanımlandığına dikkat ediniz. L1 ve L2 doğruları çemberin merkezinden geçer. Invölüt eğrisi LCONIC/4PT1SL deyimini kullanılarak tanımlanabilir. Eğri, taban çemberi ve eğrinin kesişmesindeki eğim üzerindeki dört nokta hesaplanmalıdır. Tezgah işlemi için bir APT programı yazınız. Aşağıdaki şartlar verilmiştir.

- Kesici $\frac{1}{8}$ " parmak freze
- İlerleme hızı 40 mm/dak
- Talaş derinliği 2 mm
- Devir sayısı 1500 dev/dak
- Tolerans 0.01 mm

Not: Çevresel hareket deyimini tanımlanırken okuyucu LCONIC fonksiyonuna dikkat etmelidir. Yukarıda açıklanan yolla tanımlanan ve LCONIC fonksiyonu ile temsil edilen gerçek eğri Şekil P8-2(a)'da gösterilen G eğrisinin biçimine sahip olabilir. Türetilen NC programının oldukça uzun bir dosya olduğuna da dikkat edilmelidir.

- 8.3 NC'li tezgahın otomatik bir kesici değiştiriciye sahip olduğunu ve 14 inç çaplı kesicinin diğerlerinden kesici-uzunluk kaydırma miktarının -0.785 inç olduğunu kabul ederek problem 2.5'de açıklanan parça için bir APT programı yazınız. Ayrıca, GOHOME APT deyimini ile tanımlanan ana konumun kesici değişimi için tezgah konumu olduğu farz edilmektedir. Kesici değişimi aşağıdaki iki deyimini yardımı ile programlanabilir:

```
SELCTL/n
LOADTL/n,SETOOL,t
```

burada f, n nolu kesicinin kesici-uzunluk kaydırmasıdır.

Not: GOHOME deyimini kesiciye ana konuma gitmesi emrini verir. Bölüm 13.4.1.1'e bakınız.

- 8.4 Problem 2.6'da açıklanan parçanın son işlemlerini yönlendirmesi için bir APT programı yazınız. Yüzeydeki dalgalanma yüksekliği 0.05 inç'ten az olmalıdır. Başlama ve bitiş noktaları (2, 0, 2)'dir.
- 8.5 Şekil 4-26'da gösterilen parça üç eksenli NC'li freze tezgahında işlenecektir. Bu parça için işlenmemiş malzeme 0.16 inç kalınlığındaki bir aliminyum

levhadır. Malzeme dış profile yakın olarak yerleştirilen $\frac{7}{16}$ inç çaplı üç delik ile tutulur. İç ve dış profilleri işlemek ve farklı çaptaki delikleri delmek için bir APT programı yazınız. Aşağıdaki şartlar verilmiştir:

- Kesici

KESİCİ NO.	KESİCİ	BİR NOLU KESİCİYE GÖRE KESİCİNİN UZUNLUK KAYDIRMASI (İNÇ)
1	$\frac{1}{4}$ " parmak freze	0
2	0.2" matkap	1.47
3	$\frac{3}{8}$ " matkap	2.66
4	$\frac{1}{4}$ " matkap	1.58
5	$\frac{7}{16}$ " matkap	2.90

- Devir sayısı 900 dev/dak (frezeleme)
400 dev/dak (delik delme)
 - İlerleme hızı 2.5 inç/dak (frezeleme)
2.0 inç/dak (delik delme)
 - Bitirme işlemi için verilen pay 0.02 inç
- 8.6 40 kanatlı bir türbin pervanesi (Şekil P8-6) $\frac{1}{8}$ inç çapında düz uçlu bir parmak freze kullanarak üç eksenli NC'li freze tezgahında işlenecektir. 40 kanatlı bitirme işlemini yönlendiren bir APT programı yazınız. Aşağıdaki şartlar verilmiştir:
- İlerleme hızı 1.5 inç/dak
 - Devir sayısı 2000 dev/dak
 - Tolerans 0.0002 inç
 - Başlama ve bitiş noktaları (5,4,1)
- 8.7 Problem 2.8'de açıklanan milin sağ ucunu tornalamak için bir APT programı yazınız. Koordinat sisteminin orijini, işlenmiş alım yüzeyinin merkezine ayarlıdır. APT koordinat sistemindeki başlama ve bitiş noktaları (0.5, 50)'dir. NC kodlarında vida açma çevrimini tanımlamak için APT programında INSERT deyimini kullanınız.
- 8.8 Şekil P8-8'de gösterilen kısmi emisyon pompasının pervanesi bir otomatik kesici değiştiricili üç eksenli freze tezgahında işlenecektir. İşlenmemiş malzeme önce bir tornada tornalanmakta ve ortaya çıkan profil şekilde gösterilmektedir. Tam olarak işlenmemiş bölgeleri ve delikleri frezelemek için aşağıdaki verilen şartları kullanan bir APT programı yazınız:
- Kesici

KESİCİ NO.	KESİCİ	KESİCİ UZUNLUK KAYDIRMASI (inç)
1	$\frac{1}{2}$ " 0.1 inç yarıçaplı kavisi olan parmak freze	0
2	$\frac{1}{4}$ " düz uçlu parmak freze	-0.32

- b. Devir sayısı 800 dev/dak (1. kesici için)
1500 dev/dak (2. kesici için)
- c. İlerleme hızı 3 inç/dak
- d. İzin verilen dalgalanma yüksekliği 0.005 inç
- e. Tolerans 0.001 inç
- f. Başlama ve bitiş noktaları (3,3,3)

8.9 Şekil P8-9'da gösterilen parça üç eksenli bir NC'li freze tezgahında işlenecektir. Aşağıdaki şartlar verilmektedir.

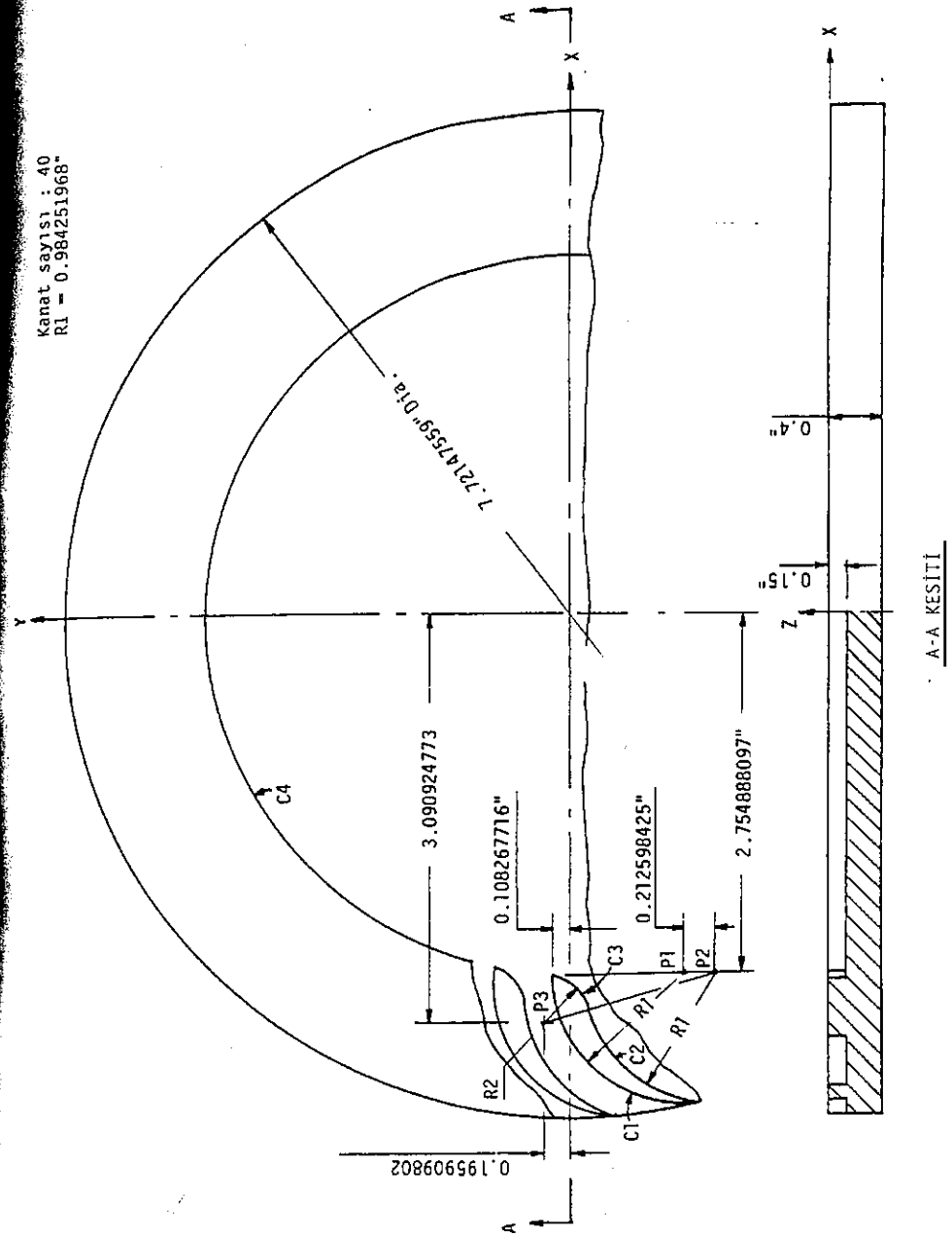
- a. İşlenmemiş parça ölçüsü: 10.2 x 5.2 x 2 inç
- b. Kesiciler, devir sayıları ve ilerleme hızları:

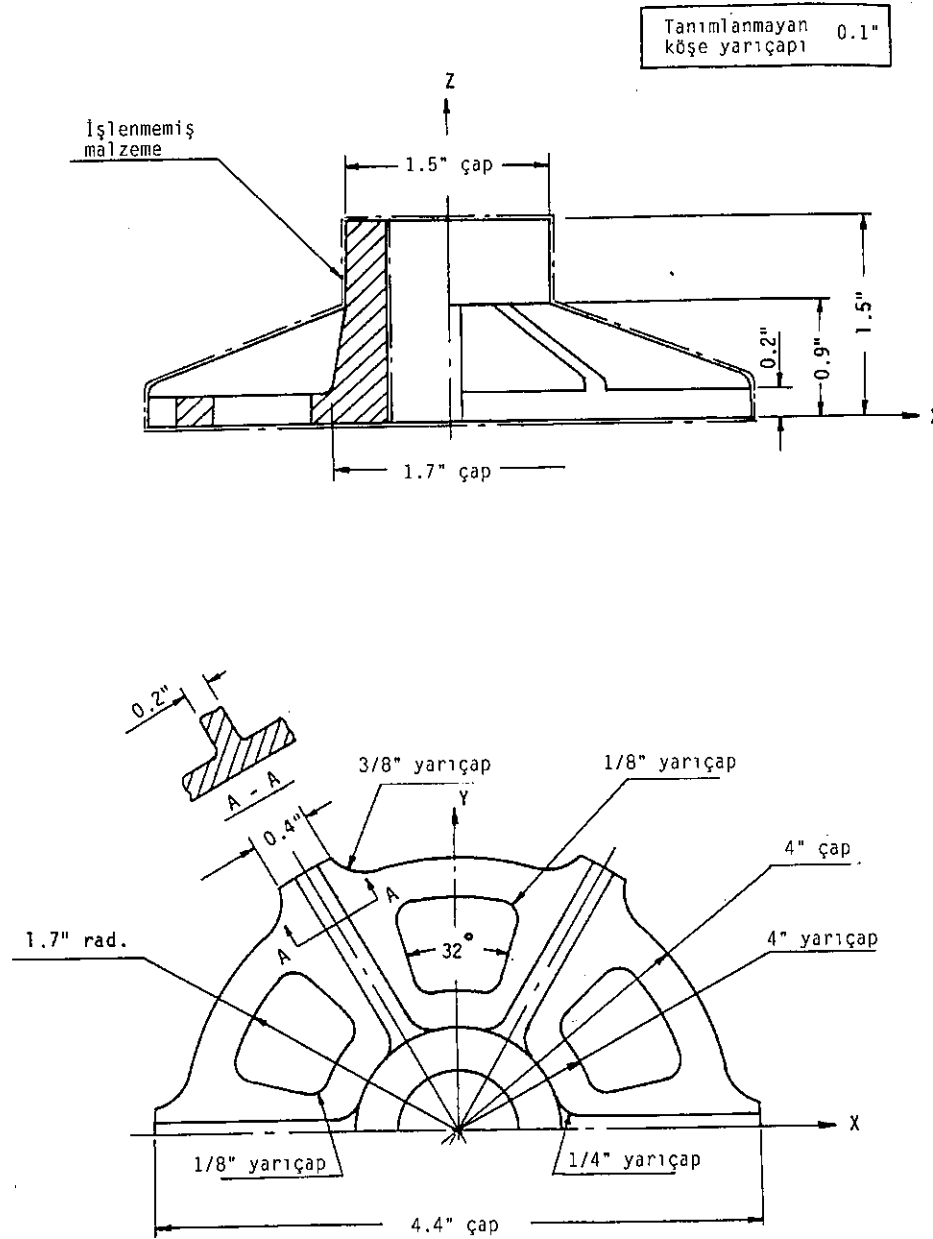
KESİCİ NO.	KESİCİ	KESİCİ-BOYU KAYMASI (inç)	İLERLEME HIZI (inç/dak)	DEVİR SAYISI (dev/dak)
1	$2\frac{1}{2}$ " çaplı alın frezeleme ve kademe frezeleme kesicisi; sert maden ucun uç yarıçapı 0.1 inç'tir.	0	5	800
2	$\frac{1}{2}$ " matkap	1.23	2.5	400
3	0.1" kavis yarıçaplı $\frac{3}{4}$ " parmak freze	-0.75	3	800

- c. Frezeleme operasyonları için maksimum talas derinliği 0.2 inç
- d. Bitirme işlemi için izin verilen pay 0.02 inç
- e. Tolerans 0.005 inç

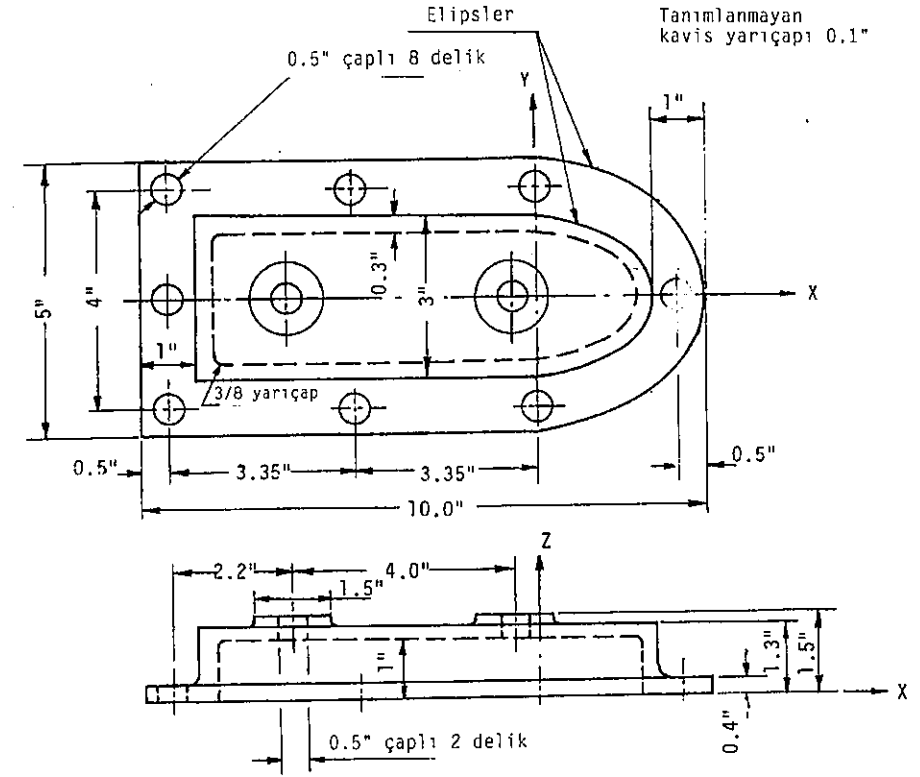
Önce, yukarıdaki z=0 parça yüzeyinin profili işlenmekte ve delikler delinmektedir. Sonra parça çevrilerek taban yüzeyi ve girintisi ölçüsünde işlenir. Bu iki işlemi yönlendiren iki APT programı yazınız.

8.10 Problem 4.8'de verilen parçayı gözününe alınız, tanımlanan S1 çok konilli yüzeyin z-en düşük noktasının, -0.5'e eşit olup olmadığını tespit etmek için bir metod tasarlayınız ve bir APT programı yazınız. POLCON deyimindeki K değerini değiştiriniz, APT programını çalıştırınız ve z-en düşük noktaya karşılık gelen koordinatını okuyunuz. Sonucu tartışınız.





Şekil P8-8



Şekil P8-9

II. PARÇA İÇİN KAYNAKLAR

1. IBM Manual SH20-1414-1: *System/370 Automatically Programmed Tool-Advanced Contouring (APT-AC) Numerical Control Processor: Program Reference Manual*, 2nd ed. Rye Brook, N.Y.: IBM Corp., 1982.
2. IBM Manual SH20-1414-2: *Automatically Programmed Tool-Advanced Contouring (APT-AC) Numerical Control Processor: Program Reference Manual*, 3rd ed. Rye Brook, N.Y.: IBM Corp., 1985.
3. Chang, Chao-Hwa, *A Guide to APT Programming*. MANE195A course notes, Manufacturing Engineering Program. Los Angeles: University of California at Los Angeles, 1985.

4. Bezier, P., *Numerical Control: Mathematics and Applications*, trans. by A.R. Forrest and A.F. Pankhurst. London: Wiley, 1972.
5. Acherkan, N.S., ed., *Handbook for Mechanical Engineers*, vol. 1, p. 255. Moscow: Mashgiz, 1961. (In Russian).
6. Korn, G. A. and Korn, R. M., *Mathematical Handbook for Scientist and Engineers*. New York: McGraw-Hill, 1968.
7. IBM Manual SH20-6640-0: *APT-AC NC Processor-Advantec Functions: Program Reference Manual*, 1st ed. Rye Book, N.Y.: IBM Corp., 1985.
8. IBM Manual SH20-1413-3: *APT-AC NC Processor: Operations Guide*, 4th ed. Rye Brook, N.Y.: IBM Corp., 1985.
9. IBM Manual SC19-6210-2: *CMS User's Guide (release 3)*, 3rd ed. Ry Brook, N.Y.: IBM Corp., 1983.

Kısım III

NC Programlarının CAD/CAM Sistemi ile Üretilmesi

Bölüm 9

CAD/CAM Sistemlerinde NC Programlama İlkeleri

II. Kısımda PT kullanımı ile SD programı durumunun detayları tartışıldı. APT'de bir kesici yolu, parça geometrisi temel alınarak oluşturulur. APT parça geometrisi ve kesici yolunun tanımlanmasında çok kolay bir çok durum sağlıyorsa da APT'de programlama büyük ölçüde kişinin parçayı değerlendirmesine ve programcının deneyimine dayanmaktadır. Diğer yandan, Bilgisayar destekli tasarım (CAD) sistemleri, parçaların çizilmesi, saklanması, ve üzerinde düzenlemeler yapılabilmesi amacıyla 20 yıldan daha uzunca süredir, sanayide geniş ölçüde kullanılmaktadır. Bir parça CAD sistemi içinde bir bilgi seti olarak tanımlanmaktadır. Eğer bir NC fonksiyonu sistem içinde yer alırsa, parçanın tanımlanabilmesi için CAD sistemi hafıza ünitesinde depolanmış bir çok bilgi NC işleme programına (Parça programı) oluşturmak için kullanılabilir.

9.1 CAD/CAM SİSTEMLERİNE GİRİŞ

CAD ve CAM sistemleri, nesnenin kafot ışın tüpü ekranında (CRT) bilgisayar kullanımı ile grafik formda hazırlanmasını ve gösterilmesini ifade eden BİLGİSAYAR GRAFİKLERİ tekniğine dayanır. Nesne, bilgisayarda matematiksel veya CAD modeli diye isimlendirilen matematiksel veya geometrik bilgi formunda saklanır. Bilgisayarı işleme hızının üstünlüğü ile herhangi biri, modeli bilgisayar etkileşimi ile daha iyi bir tanımlama için ele alabilir ve analiz edip değerlendirile-

bilir. Nesnenin grafik olarak tanımlanması için ihtiyaç duyulan fonksiyon ve bilgilere ilaveten test etme, üretim ve yönetimle ilgili bilgileri de bilgisayar sistemine ekleyebilir.

Basit bir CAD sistemi kullanıcıya grafik nesneyi oluşturma, ekranda görme, ele alma ve üzerinde değişiklikler yapma imkânını verir. Bir çizicinin kullanımı ile elde edilen kalitede cisim oluşturma, herhangi bir biçimde saklama (kaset ve disket) ve çabucak değişiklik yapabilmek için geri çağırma imkânı sağlar.

Bir CAD sistemi ayrıca, ağırlık merkezi, atalet momenti, kinematik analiz ve sonlu elemanlar analizi vs. hesaplamalar gibi tasarım işlemlerini içerebilir. Böylece çeşitli tasarım seçenekleri ekranda denenip test edilebilir ve en uygun tasarıma daha kolay bir şekilde ulaşılabilir.

CAD sistemi, sayısal denetim fonksiyonu sağlayan bir yazılımın kullanımı ile daha da güzelleştirilebilir ve bu yazılım tezgah işlemleri ve parça profiline dayanan kesici yolunun ekranda tanımlanabilmesine imkân sağlar.* CAD veya diğer yüksek düzey NC programlama dillerinde hazırlanmış ve kesici hareketi ile tezgah işlem sırasını tanımlayan bir parça programı, otomatik olarak oluşturulabilir. Ayrıca işlem planlama ve üretim simülasyonu imkânı veren yazılımları içeren geliştirilmiş CAD sistemleri de vardır. Parçanın sınıflanması, işlem planlama ve diğer işlemler için gerekli bilgiyi içeren yoğun bilgi tabanı ve geometrik modeller esasına dayanan CAD sistemleri, planlama ve üretim işlemlerinin simülasyonunda üretim mühendisini yönlendirebilir. Geliştirilmiş NC özellikli CAD sistemleri, kullanıcıyı son işlemcinin oluşturulmasında yönlendirecek bazı programları da içermektedir. Hem tasarım hem de üretimi destekleyen bu tip geliştirilmiş sistemler, CAD/CAM SİSTEMLERİ olarak isimlendirilir.

Çok yönlülük ve güçlerine rağmen CAD veya CAD/CAM sistemleri elbette tasarım ve analiz işini tam anlamıyla otomatik olarak yapmayabilir. Bu sistemlerin amacı oldukça karmaşık tasarım işlemi süresince mühendise yol göstermektir. Böylece tasarımcı ve CAD sistemi herbirinin en iyi özelliklerinin birleşiminden oluşmuş bir takım gibi ele alınır. Kişi bir tasarımı oluşturur, sisteme yerleştirir, dener, düzenler, bilgide değişiklikler yapar ve kararlar verir. CAD sistemi ile gerekli sayısal analiz ve organizeleri yapar, saklar, nakleder ve bilgiyi girer. Her bir eleman yapabileceğinin en iyisini yapar. Bilgisayar destekli tasarımın sonucu, işlenecek parçanın sayısal formda tanımlanan geometrik veri tabanıdır (veya CAD modeli).** Sonra CAM yazılımı, geometrik veri tabanında parçanın üretimi için arzu edilen bilgiyi oluşturmak için kullanılır. Bu işlem, süresince, tasarımcı veya kullanıcı CAD'daki aynı rolü oynamaktadır. Örneğin NC programında çalışırken tasarımcı başlama noktasını, işlem sırasını, tezgah özelliklerini ve kesici yolunu

* Aslında CADAM gibi ilk CAD sistemleri bu tip NC fonksiyonunu oluşturmak için tasarlanmıştır.

** Veri tabanı olarak isimlendirilmesine rağmen, bu geometrik bilgi genellikle kelimenin tam anlamıyla gerçek veri tabanından çok, özel dosyalama sistemidir.

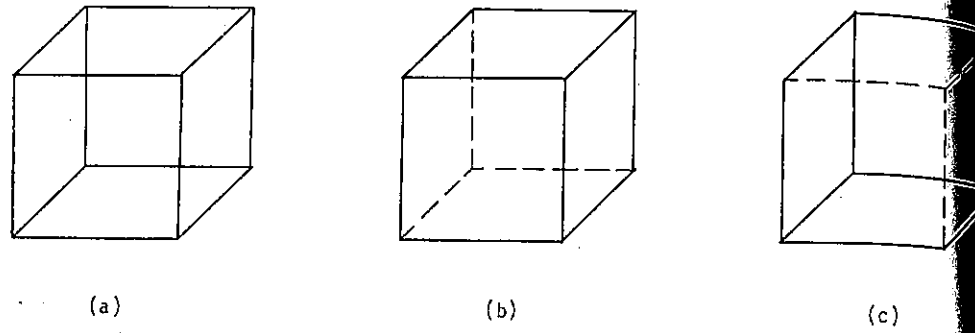
hazırlamalıdır. CAD/CAM sistemi, gerekli matematiksel hesaplamaları çıkarır, hesaplanan sonuçları NC programına tasarımcı tarafından girilen bilgileri çevirir.

Son yıllarda gözlenen eğilim, CAD/CAM sistemine tasarımcı tarafından verilen çok fazla fonksiyonunun birleştirilmesi olmuştur. Bunun sebebi, parçaların, tezgahların ve işlemlerin çok karmaşıklaşması, buna bağlı olarak NC programcılarının, üretim planlayıcılarının ve deneyimli insanların hızla azalması olmuştur. Deneyim ve alanda uzmanlaşmış programcı bilgisini birleştiren yapay zeka programları CAD/CAM sistemleri içinde tasarlanmaktadır.

Varolan CAD sistemleri üç temel biçime ayrılabilir: mikroişlemci (mikrobilgisayar) işletici sistemler, mini bilgisayar işleticili sistemler, terminalli sistemler. Mikrobilgisayara dayalı sistemler en küçük ve en ucuz olanlardır. Normal olarak basit çizim işlemleri yapmaya imkân verir ve bazı sınırlı analitik fonksiyonları vardır. Bu tip sistemler genellikle çizim işlemleri için kullanılır ve işleme hızları bu üç grup içinde en yavaş olanlardır. Minibilgisayar işleticili sistemler çizim yazılımına ilaveten, tasarım, üretim veya her ikisinde de kullanılan, çeşitli analitik fonksiyonları içeren programlardan ibarettir. Minibilgisayar grafik sistemleri, tasarım üretim ve üretim yönetimi için arzu edilen geniş veri tabanı hazırlayabilen ve karmaşık özellikli parçaların üretimi için büyük ölçüde bilgi işleminin ihtiyaç duyulduğu durumlarda gereklidir.

Bir CAD/CAM sistemi bilgisayar hafızasındaki oluşturulmuş giziksel bir nesnenin geometrik modelinde çalışır. Parçanın geometrik modeli ile tanımlanan işlem GEOMETRİK MODELLEME olarak isimlendirilir. Tel çerçeve, yüzey ve katı olmak üzere üç çeşit üç boyutlu modeller CAD/CAM sistemlerinde kullanılır. Bir tel çerçeve modeli parçayı kenar ve köşeleri ile tanımlar. Model, bilgisayarda parçanın kenar ve köşelerini tanımlayan noktaların, doğruların, eğrilerin bir listesi olarak saklanır. Tel çerçeve model, bir parça yada nesnenin üç boyutlu uzayda tanımlanması için bilgisayar grafiklerinde kullanılan en eski ve en basit metoddur. Yine de şu anda kullanılan bir çok sistem bu metoda dayanmaktadır. Tel çerçeve modelleme, diğer üç boyutlu modelleme metodlarına göre en az hafıza ve çok az işleme zamanı gerektiren bir metoddur. Bununla beraber, parçaların hacim ve yüzeylerine bağlı herhangi bir bilgi vermektedir. Bundan dolayı bir tel çerçeve model genellikle yetersizdir. Örneğin bir tel kübün tel çerçeve modeli CRT ekranında Şekil 9-1(a)'daki gibi görüntülenmektedir. Bu şekle göre parça Şekil 9-1(b) veya 91(c)'deki gibi de düşünülebilir. Sonuç olarak, tel çerçeve model bir nesnenin tanımlanmasında yetersiz kalmaktadır. Bununla ötesinde sınır yüzeylerin, kesişme kenarlarını SOLELY tanımlar ve hacim olarak değişen yüzeyleri tanımlamaktan uzaktır. Bunun sonucu olarak NC ürün için kullanılan kontrol elemanları (parça tahrik (sürücü) ve kontrol yüzeyleri) kesicinin herbir hareketi için doğrular ve eğriler şeklinde tanımlanmalıdır.

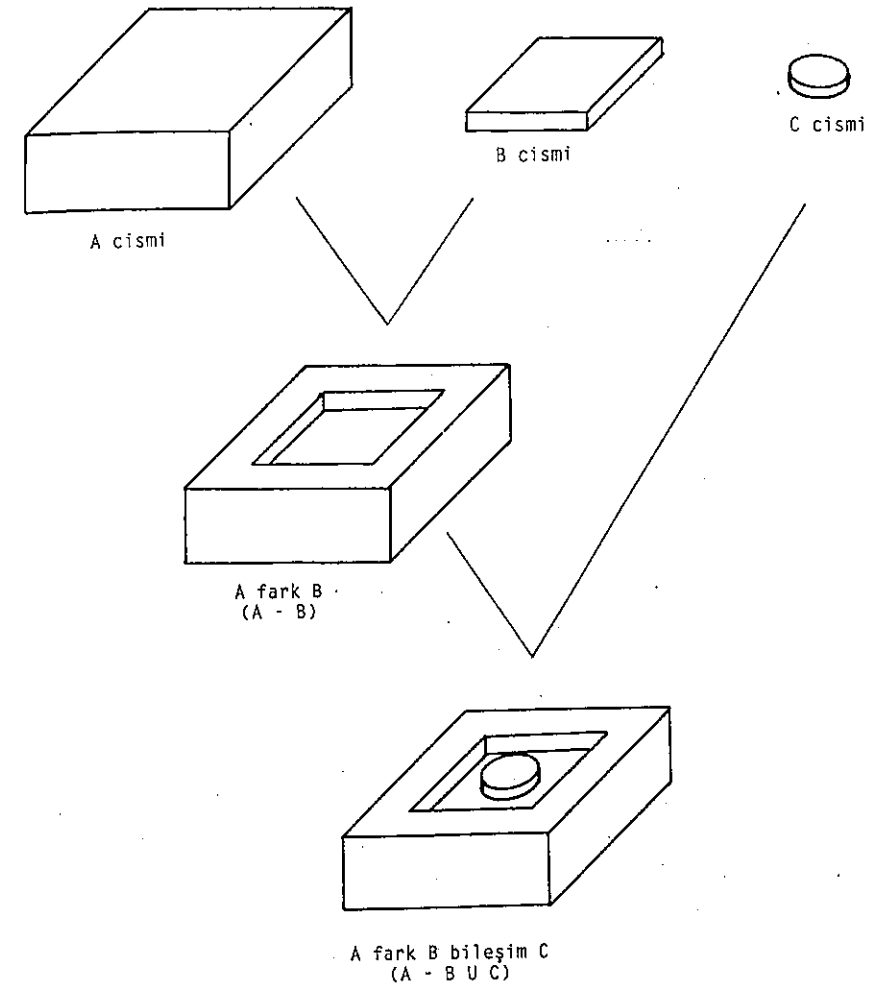
Yüzey modelleme de, bir nesne veya parça, sınır yüzeylerini analitik olarak



Şekil 9-1 Küpün (a) ve muhtemel cisimlerin (b v c) (modelle temsil edilen) tel çerçeve modeli.

tanımlanan bilgi demeti ile ifade edilir. Bir parçanın yüzey modeli tel çerçevedeki sınır yüzeylerinin tanımlanması ile oluşturulabilir. Bunlar, düzlemler, silindirlere ve koniler veya karmaşık eğrilere sahip yüzeyler gibi basit yüzeyler olabilir. Sistem geometrik model ve bilgileri depolamak için bir merkezi veri tabanı kullanır. Böylece tasarımcı ve üreticilerin geometrik ve alfasayısal bilgiyi paylaşması temin edilir. İki ve üç boyutlu eksenli NC programları değişik görünüşlerde, parça geometrisine sahip olarak oluşturulabilir. Kesici hareketi ekranda görüntülenir, ve değerlendirilmesi için tekrar ek alınabilir.

Yüzeylerin tanımlanması, görünümünü etkileyen renk, pürüzlülük, şeffaflık, gölgeleme gibi özelliklerin katılması ile artırılabilir. Bir yüzey model, parça hakkında gözle görülebilir sayıda detayları verir ve herhangi bir yüzeye bağlı olarak kesicinin çevresel işleme hareketini oluşturmada kullanılabilir. Böylece, yüzey model tel çerçeve modeldeki çizgi veya eğrilerden çok uzaysal tanımlanması için kullanılır. Böylece sayısal denetimli kesici yolunun oluşturulması hayli basitleştirilmiş olur. Hacim, ağırlık ve atalet momenti gibi analiz ve uygulamalar için gerekli bilgilerin eldesi için katı modelleme kullanılmalıdır. Yukarıda tanımlanan iki yaklaşım karşılaştırılırsa, katı modelleme nesnenin köşe, kenar ve sınır yüzeylerine bağlı olarak katı biçimini tanımlamayı mümkün kılar. Katı modellemede kullanılan iki temel tanımlama tekniği vardır: Yapısal Katı Geometrisi (CSG), Sınır tanımlaması (B-rep) CSG. Düzlem, küp, silindir, sıfır ve koni gibi basit katları Boolean işlemleri boyunca bir nesnenin tanımlanması için kullanır. (Şekil 9-2). Bilgisayarda, nesne ağaç (tree) diye isimlendirilen bir grup bilgi ile tanımlanır. Bu ağaç yapıda temel nesnelerin yeri ve pozisyonları, onlara uygulanan dönüşümler ve bu temel nesnelerin ağaçlar yardımıyla nasıl bir araya getirildiğini tanımlayan Boolean işlemleri bulunmaktadır. Sınır tanımlamasında, bir nesne onun ilgili sınırı ile birlikte sınırın hangi yanında katının yerleştirildiğini işaret eden bilgi ile birlikte



Şekil 9-2 Basit katılarla başlayan, Boolean operatörü kullanarak bir cismin tanımlanması.

tanımlanır. Katı model bilgisayarda yüzeylerin sınırladığı bir hacmin hesaplanması gibi matematiksel olarak ifade edilir. Model ayrıca, parçanın tanımlanması için yeterli tam bilgi ve mühendislik analizlerinde hesaplamalar için kullanılacak atalet momenti, kütle gibi bilgileri de içermektedir. Parçanın geometrik özelliklerine bağlı veya doğrudan CAD modelinden NC programının otomatik oluşumu daima nesnenin hacminin gözönünde tutulması zorunluluğunu göstermektedir (Bölüm 9-2'ye bakınız). Bundan dolayı parça katı model ile yeterli şekilde tanımlanabilir.

Daha detaylı bilgi bu kitabın amacı dışındadır. İlgilenen okuyucular Besant ve Liu³ ve Mortenson'un eserlerinden detaylı bilgi alabilir.

Tasarım çizimlerine dayalı NC programları oluşturmada özelliği olan bir çok CAD/CAM sistemi 1960'ların sonunda geliştirildi. CADAM INC. tarafından geliştirilip IBM tarafından piyasaya sürülen CADAM, Fransız Dassault System Co. tarafından geliştirilen ve IBM tarafından ticari kullanıma sunulan CATIA ve Computer Vision Corp. tarafından geliştirilen CDS-CAD/CAM sistemi en yaygın kullanım alanı bulmuş CAD/CAM sistemleridir. NC program oluşturmada mümkün kulan, tasarım ve üretim özellikli birçok CAD/CAM programlarını piyasaya sunan pek çok firma mevcuttur. Bunların detaylı listesi 5. ve 6. referanslarda bulunmaktadır.

CATIA, IBM tarafından desteklenmiş etkileşimli terminal tabanlı bir CAD/CAM grafik sistemidir. Sistem grafik ve alfasayısal olmak üzere iki konsülden meydana gelmiştir. Bu özellik, kullanıcıya aynı anda çizimi ve tasarım bilgisini okuma imkanı verir. Sistem katı ve yüzey modelleme, üç boyutlu tasarım ve çizim ortamları için tümleşik bir veri tabanı sağlar. Bir, iki, üç, dört veya beş eksenli NC işlemi için hazırlanan APT parça programı geometrik modelden oluşturulabilir. Kesici yolu yerleşme konumlarındaki hataları gösteren işaret ve sembollerle görüntülenebilir. Ayrıca sistem CADAM ile ilişki kurup, CADAM veri tabanından bilgi çıkarımı sağlayabilir.

9.2 BİR CAD MODELİNE BAĞLI NC PROGRAMI OLUŞUMUNUN İLKELERİ

CAD modelinden kesici yolu oluşturma birçok yolu vardır. Tel çerçeve modelde, tanımlanan geometrik elemanlar, nokta, doğru ve eğrilerdir (Bir geometrik elemanın kenar ve köşeleri). Bundan dolayı, tel çerçeve modele dayalı CAD/CAM sisteminde, çevresel işlemeli kesici yolu, tahrik, kontrol ve parça yüzeylerini doğru belirleyen eğrilerle tanımlanır. APT-AC NC işlemcisinde kullanılan ile benzer algoritma çevresel işlemede kesici pozisyonlarını hesaplamak için kullanılır.

Yüzeyler CAD/CAM sisteminde tanımlandığı zaman, profilin oluşturulmasında kullanılacak kesici yolu otomatik olarak hazırlanır. Örneğin (Şekil 9.3) bir yüzey aşağıdaki vektörel yazılım ile tanımlanabilir.

$$\mathbf{r} = \mathbf{r}(u, v)$$

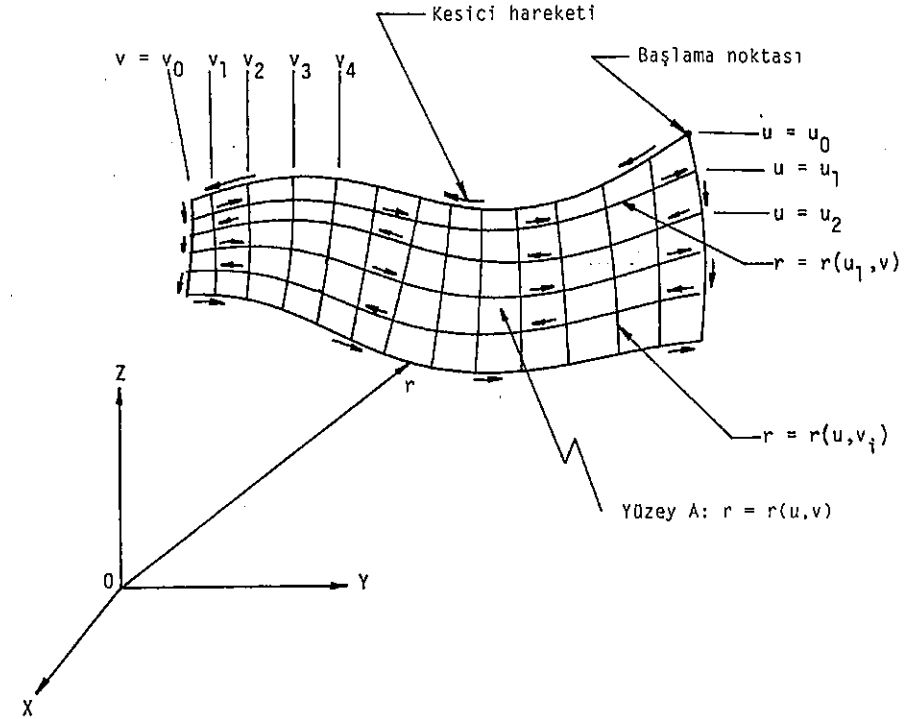
r: Yerleşim vektörü

u, v: A yüzeyinin parametreleri

Eğer parametrelerden biri sabit tutulursa ve;

$u = u_1$ veya $v = v_1$ olursa, vektör eşitliği;

$$\mathbf{r} = \mathbf{r}(u_1, v)$$



Şekil 9-3 $\mathbf{r} = \mathbf{r}(u, v)$ vektör eşitliği ile tanımlanan bir yüzey ve yüzeyin oluşturulması için parametrik eğriler boyunca kesici hareketi.

veya

$$\mathbf{r} = \mathbf{r}(u, v_1)$$

şeklini alır. Bu yazılım, bir yüzey üzerindeki bir eğriyi tanımlar ve A yüzeyinin PARAMETRİK EĞRİSİ ismini alır. Yüzeyin profili için kesici hareketi, kesicinin parametrik eğri boyunca hareketini temin için tanımlanır. Parametrik eğri boyunca kesicinin başlangıç ve bitiş noktaları diğer bazı eşitlikler yardımıyla tanımlanan yüzey sınırları ile hazırlanır. Yüzey profilinin başlangıç noktası ayrıca seçilebilir. Böylece kesicinin profil açısı tam olarak tanımlanabilir. Kesici parametrik eğri yüzeyinde zigzag yapacak şekilde hareket eder (Şekil 9.3) ve "u" ile "v" artış değerleri kesici geometrisi ve yüzeyde sola izin verilen sapma yüksekliği ile sağlanır. Yukarıda tanımlanan teknikler ışığında bir algoritma hazırlanır ve CAD/CAM sistemi içinde yer alır. Bir yüzey ve onun sınırlarının tanımlanması ile yüzey profili için kesici yolu, kesici geometrisi, izin verilen sapma yüksekliği, tolerans ve kesici hareketinin başlangıç noktasının özelliklerinden sonra otomatik

olarak oluşturulabilir. Bu temel prensiplere göre hazırlanmış CATIA gibi ticari CAD/CAM sistemleri, kesici hareketini otomatik olarak hazırlayan ve kullanıcının bu hareket için adım adım bilgi girişini gereksiz kılan fonksiyonları içermektedir.

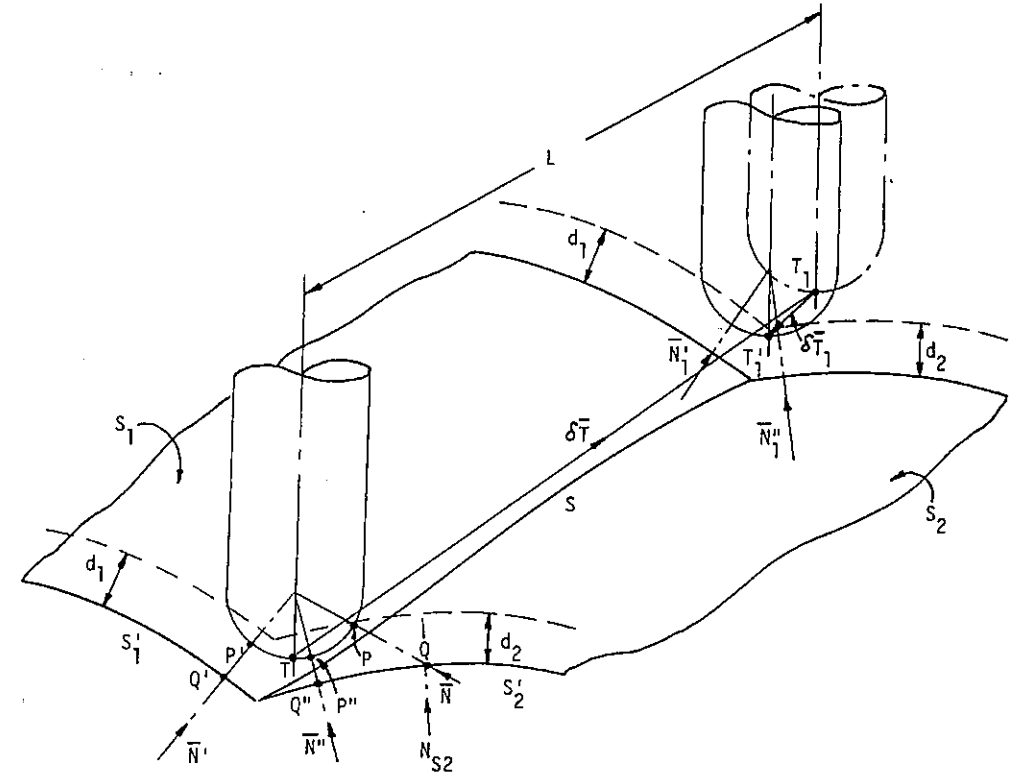
Genelde işlenmiş bir parça, kaba ölçülerdeki bir parçadan fazla parçanın çıkarılması şeklinde kabul edilebilir. Kesici hareketi veya parçanın kesilmesi, parçanın resme göre son hali için fazlalık olan hacmin çıkarılması anlamına gelmektedir. Kesici hareketini iki parçaya ayırmak mümkündür: Birincisi kesici profilini tam izlemesi gerekmeyen, fakat iş üzerindeki fazla talaşın alınmasını temin eden "Kaba İşleme"; diğeri ise kesicinin sınır yüzeylerini veya parça profilini tanımlayan "İnce İşleme"dir. Bir iş için kesici yolunu hazırlamak amacıyla bir algoritma tasarlanabilir ve parçanın katı modeline bağlı olarak bilgisayar grafik sisteminde yer alır. Böylelikle, NC kesici yolu, izin verilen sapma yüksekliği, tolerans, başlama ve bitiş noktaları, gerekli işleme özelliklerinin tanımlanmasından sonra, kesici, iş parçası ve yarı mamul malzemenin katı modellerine bağlı olarak otomatik olarak oluşturulur. Bağlama kalıplarının katı tanımlamaları gibi ilave bilgiler, kesicinin kalıplara çarpmasını önlemek amacıyla sisteme ilave edilebilir. Katı modelleme tekniklerinin parçanın tam ve yeterli tanımlanmasını yapmasından bu yana, CAM sistemleri için en ideal modülün NC programının otomatik olarak hazırlanması olmuştur.

9.3 CAD/CAM SİSTEMİ İÇİNDE KESİCİ HAREKETİNİN OLUŞTURULMASI

Bu bölümde bir CAD/CAM sistemi içinde kesici hareketlerinin nasıl hazırlandığı açıklanmaktadır. Tartışılan kesici hareketi, çevresel hareketin tek adımını ve yüzey profili için kesici yolunu ifade etmektedir. Noktadan noktaya hareket, kesicinin bulunacağı son noktanın koordinatları ile basit olarak hazırlanır. Bu metodun açıklanmasına gerek duyulmamıştır. Bir çok CAD/CAM sistemi temel işleme programının oluşturulması için CAD modelinin geometrik bilgisini kullanmaktadır. Sonra NC işlemcisi ve NC parça programını oluşturacak uygun son işlemciyi kullanır.

9.3.1 Basit Çevresel Hareket

Çevresel hareket parça profilini izlemekte ve kesicinin profilden sapması önceden hazırlanmış toleranslar dahilinde olmaktadır. Bir çevresel hareket bir seri doğrusal enterpolasyon hareket adımından meydana gelmektedir (Bölüm 5.9'a bakınız). Tek bir hareket adımını tanımlayabilmek için iki hareket elemanına ihtiyaç vardır: Hareket doğrultusu ve mesafe (boy). Bu ifadeler tahrik ve parça yüzeyleri bu yüzeylere ait toleranslar ile hazırlanmaktadır. Kontrol yüzeyi son

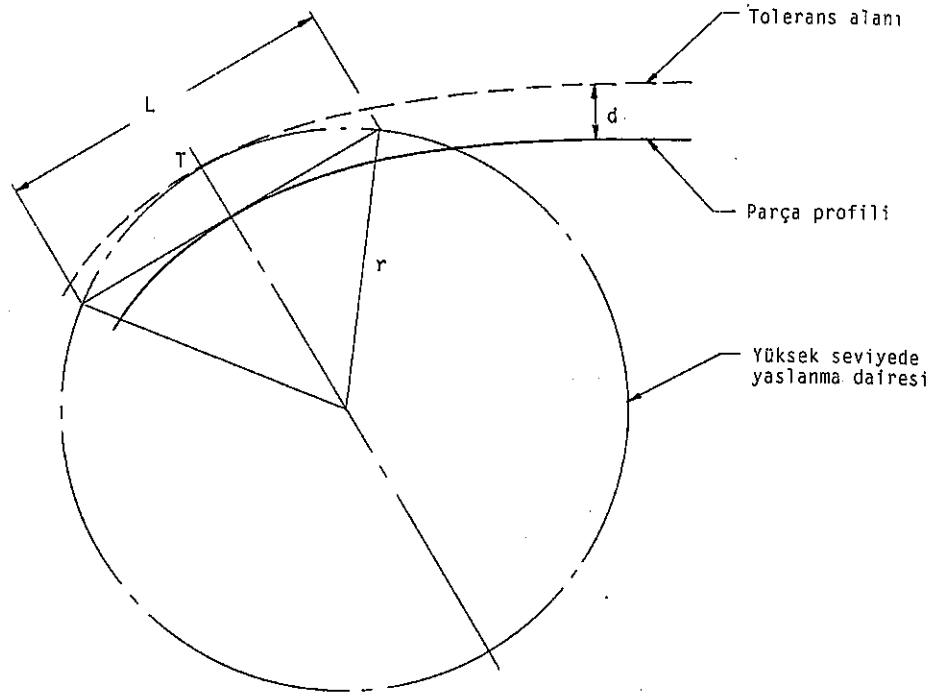


Şekil 9-4 Doğrusal enterpolasyon hareket adımının tanımlanması: d_1 ve d_2 : S_1 ve S_2 yüzeyleri için toleranslar.

hareket adımında bitiş noktasını tanımlamak için kullanılır. Aşağıdaki algoritma APT III ve HAPT-3D işlemcilerinde kullanılanlardan biridir⁷⁻⁸.

Kesicinin çevresel işlemeye hazır olarak T noktasında beklenildiğini varsayalım (Şekil 9-4). Kesici profili ile iki kontrol yüzeyi arasındaki mesafeler S_1 ve S_2 ($P'Q'$ ve $P''Q''$), d_1 ve d_2 toleranslarından daha azdır. \bar{N} ' nün kesici profili ve tahrik yüzeyi S_1 'e; \bar{N}'' nün parça yüzeyi S_2 ve kesici profiline ortak normal olduğunu kabullenelim. S_1' ve S_2' ; S_1 ve S_2 yüzeyli \bar{N}' ve \bar{N}'' ile tanımlanan düzlemin arakesitleridir. Buna göre doğrusal enterpolasyon hareket adımının doğrultusu $\delta\bar{T}$ hem \bar{N}' hem de \bar{N}'' normallerine dik olmalıdır.

\bar{N}' ve \bar{N}'' normalleri aşağıda açıklanan⁷ prensiplere bağlı olarak iteratif olarak hesaplanır. Kesici konumu T için kesici üzerinde rastgele bir P noktası seçilir. Kesici profiline normal \bar{N} , S_2 yüzeyi ve \bar{N} nin kesişme noktası Q ve Q noktasında S_2 yüzeyine normal (\bar{N}_{S_2}) hesaplanır. Sonra \bar{N}_{S_2} ve \bar{N} arasındaki açı bulunur, ve



Şekil 9-5 d tolerans değerine ve r ağırlık yarıçapına bağlı doğrusal enterpolasyon hareketinin, L adım uzunluğunun tanımlanması.

$$L = 2 [d \cdot (2r - d)]^{1/2}$$

önceden tanımlanmış küçük açı değeri ile karşılaştırılır. Örnek olarak APT'de karşılaştırılan değeri 0.01 radyan olarak verebiliriz. Eğer hesaplanan açı bu değerden küçük ise hesaplanan normal \bar{N} S_2 yüzeyi ve kesici profili için ortak normal olarak kabul edilir. Aksi takdirde Newton-Raphson metoduna göre Q noktasında S_2 yüzeyinin kavisinin kullanımı ve yeni bir Q noktasının tahmini yapılır^{7*}. Bu işlem, elde edilen (\bar{N}'') istenen hassasiyete ulaşıncaya kadar tekrar edilir. Ortak teğet, \bar{N} 'de benzer şekilde hazırlanır. Şekil 9-4'deki hareket doğrultusu $\delta\bar{T}$, \bar{N} ve \bar{N}'' vektörlerine dik olmalıdır. Böylece, hareket doğrultusu aşağıdaki vektörel tanımlama ile hazırlanabilir.

$$\bar{N}'' \times \bar{N}' / |\bar{N}'' \times \bar{N}'|$$

L mesafesi veya hareket adımının son noktası, öncelikle tolerans değeri ve T

* Newton Raphson metodu, polinomial eşitliklerin herhangi bir hassasiyette nümerik (sayısal) çözümünü sağlayan basit bir hesaplama işlemidir.

noktasındaki S eğrisinin arakesitinin kavis yarıçapına bağlı olarak tahmin edilir (Şekil 9-5). Böylece son noktanın ilk kesin olmayan pozisyonu T_1 olarak belirlenir (Şekil 9-4). Bununla beraber, kesicinin bu yerleşimi S_1 ve S_2 yüzeyleri için d_1 ve d_2 toleranslarını aşabilir. Bunun için T_1 'den $\delta\bar{T}_1$ kadar bir hareketle düzeltme yapılmalıdır. Bu doğrultma hareketinin son konumu, T_1 ' aşağıdaki üç durumda hazırlanır.

1. S_1 yüzeyi ile kesici profili arasındaki mesafe d_1 'den küçük veya eşit olmalıdır.
2. S_2 yüzeyi ve kesici profili arasındaki mesafe d_2 'den küçük veya d_2 'ye eşit olmalıdır.
3. $\delta\bar{T}_1$ artış miktarı \bar{N}_1' ve \bar{N}_1'' normali ile tanımlanmış düzende olmalıdır.

Bu üç durum aşağıdaki üç vektörel eşitliği sağlar⁷:

$$\bar{N}_1' \cdot \delta\bar{T}_1 = D_1 \leq d_1 \quad (9.2)$$

$$\bar{N}_1'' \cdot \delta\bar{T}_1 = D_2 \leq d_2 \quad (9.3)$$

ve

$$(\bar{N}_1'' \times \bar{N}_1') \cdot \delta\bar{T}_1 = 0 \quad (9.4)$$

\bar{N}_1' ve \bar{N}_1'' yukarıda sözü edilen metodla hazırlanır, ve $\delta\bar{T}_1$ değeri D_1 ve D_2 mesafelerinden bir tolerans değerine eşit oluncaya kadar ve diğer tolerans sınırı içindeki bir değere eşitleninceye kadar ayarlama yapılır. Hesaplama, bilgisayar tarafından otomatik olarak yapılan iteratif bir işlemdir. Sonuç olarak, kesicinin son noktası T_1' olarak hazırlanır. Sonra hareket adımının gerçek uzunluğu bulunur ve hesaplama, tolerans sınırı içinde TT_1' hareket adımını sağlayan herhangi bir nokta sağlanıncaya kadar yapılır. Eğer bu sağlanırsa, hareket adımının uzunluğu düşünülmelidir. Aksi takdirde, T_1' noktası kabul edilir. Bir sonraki hedef noktanın hesaplaması T_1' noktasının hazırlanmasından sonra başlar.

Bu çevresel hareketi, sonunda, kontrol yüzeyi, S_3 (Şekil 9-4'te gösterilmiştir) son hareket adımının son noktasının hazırlanması için kullanılmalıdır. Buna göre 9.4 eşitliği aşağıdaki ile yer değiştirmelidir.

$$\bar{N}_n \cdot \delta\bar{T}_n = 0$$

Burada:

\bar{N}_n : Son hareket adımının son noktasındaki S_3 kontrol yüzeyine olan normal.

$\delta\bar{T}_n$: Son hareket adımının son noktasındaki ayar hareketidir.

Yukarıda sözü edilen yonteme bağlı olarak bir algoritma tasarlanabilir, ve çevresel harekette kesici konumlarını hesaplamak için CAD/CAM sistemi içine yerleştirilebilir. Çevresel hareketi tanımlayan birçok APT GOTO deyimleri, bu hesaplanmış konumlara göre oluşturulur.

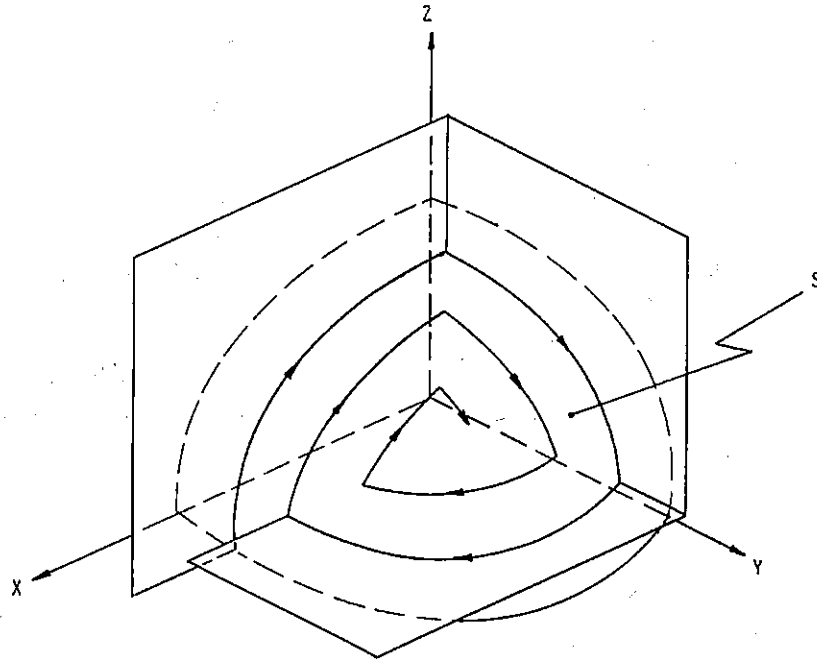
9.3.2 Bir Yüzeyin Oluşturulması için Kesici Hareketinin Hazırlanması

Bir yüzeyin oluşturulması için gereken kesici hareketi, tolerans değerinden küçük veya eşit değerde, yüzey ve kesici arasındaki mesafenin oluşturduğu yüzeyde kesicinin hareketinden meydana gelir. Kesici hareketi, yüzey eğrilerinin kesişme noktasını kontrol düzlemlerine paralel düzlemi (Şekil 9-6) veya (Şekil 9-3)'te gösterilen parametrik eğriyi takip eder. Genellikle, tek bir parametrik oluşumun tanımlanmasına göre kontrol yüzeyler gibi parametrik eğrilerin kullanımı daha basittir. Bunun yapılması ile hesaplama işlemleri bir hayli basitleştirilmiştir.

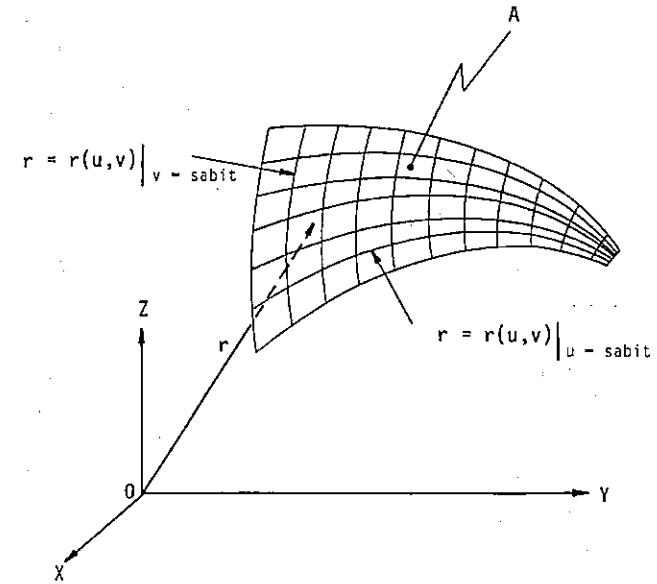
Bir temel hareket doğrultusu, parametrik eğri ile tanımlanan kesici hareketi için seçilmelidir. Şekil 9-3 eğri boyunca olan temel hareket doğrultusunu gösterir.

$$\mathbf{r} = \mathbf{r}(u, v) \Big|_{u=\text{sabit}}$$

İzin verilen dalgalanma yüksekliği ile hazırlanmış ilerleme hareketi eğri boyunca



Şekil 9-6 S yüzeyinin oluşturulmasında kesici hareketi. Kesici hareketinin yolu, S yüzeyi ile koordinat düzlemlerine paralel düzlemlerin kesişme eğrileriyle tanımlanır.



Şekil 9-7 Temel harekete kılavuzluk edecek $\mathbf{r} = \mathbf{r}(u, v) \Big|_{v=\text{sabit}}$ parametrik yüzeyin seçimi.

$$\mathbf{r} = \mathbf{r}(u, v) \Big|_{v=\text{sabit}}$$

eşittir.

Bazı durumlarda, temel hareket eğrisi gibi parametrik eğrilerin seçimi, üretim zamanını önemli ölçüde etkilemektedir. Şekil 9-7'de A yüzeyi sağ tarafta çok dardır. Eğri boyunca temel hareket aşağıdaki eşitlikle ifade edilirse, sağa doğru yapılan tekrarlı hareket zaman kaybına neden olur.

$$\mathbf{r} = \mathbf{r}(u, v) \Big|_{u=\text{sabit}}$$

bundan dolayı;

$$\mathbf{r} = \mathbf{r}(u, v) \Big|_{v=\text{sabit}}$$

ile ifade edilen, aynı zamanda temel hareketin kontrol yüzeyi görevini yerine getiren eğri seçilmelidir.

Temel hareket yönü bir defa tanımlandığında, profili oluşturmak için gerekli hareketin hazırlanması Bölüm 9.3.1'de tanımlanan tekniğin kullanımı ile basitleştirilir ve her bir profillemeye hareket adımı için kesicinin konumlandırma hesapları azaltılır. Böylece, profillemeye (işleme) hareketi, başlangıç noktasının belirlenmesinden sonra tam olarak tanımlanabilir.

Hesaplama işlemleri yukarıda sözü edilen metoda bağlı olarak tasarlanır ve herhangi bir (rastgele) yüzey için kesici hareketlerinin APT yazılımı ile tanımlanması, belirli bir yüzey, kesici geometrisi, başlama noktası, toleranslar, ilerleme miktarı ve izin verilen dalgalanma yüksekliğine bağlı olarak otomatik olarak oluşturulabilir.

9.4 NC PARÇA PROGRAMLARININ DOĞRUDAN CAD MODELİNDEN OTOMATİK OLUŞUMUNA BAĞLI DİKKAT EDİLMESİ GEREKEN HUSUSLAR

CAD/CAM sistemlerinin gelişimindeki son yıllardaki eğilimlerden biri, NC parça programlarının doğrudan BDT modelinden otomatik olarak oluşturulmasıdır. Gerçi etkileşimli CAD/CAM sistemleri program yazımındaki zamanı azaltmaktadır. Fakat bu etki belirli durumlarda büyültülecek bir olay değildir (Bölüm 10'a bakınız). Bunun yanında iyi tasarlanmış bir işleme planı, işleme adımlarının etkileşimli bir şekilde hazırlanabilmesi için önceden tasarlanmalıdır. NC işleme programının doğrudan CAD modelinden otomatik olarak oluşumu OTOMATİK İŞLEM olarak tanımlanır. Bu işlemle işleme stratejisi tasarlanır, kesici yolu hazırlanır, ve CAD/CAM sisteminin veri tabanında bulunan üretim ve parça ile ilgili geometrik özelliklere bağlı olarak NC programı oluşturulur. Son yıllarda önemli sayılabilecek girişimler, bu tip sistemlerin geliştirilmesi üzerine yapılmaktadır⁹⁻¹². Bununla beraber, bu alandaki araştırmalar hala başlangıç aşamasındadır. Aşağıdaki açıklamaların amacı temel problemleri ve yaklaşımı basitçe tanıtmaktır.

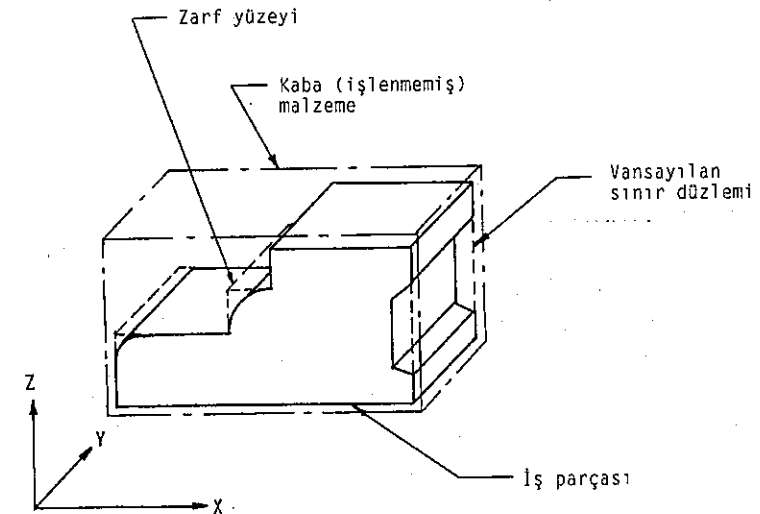
Genellikle tam bir tezgah işlemi kaba ve ince işleme olmak üzere iki kısma ayrılabilir. Kesici hareketi, kaba ölçülerdeki parça üzerinden fazla talaş kaldırmak için parça geometrisi ve kaba ölçülerdeki malzeme ölçülerinden hazırlanır. Sonra kesici hareketi, bu hacim, kullanımdaki bağlama elemanlarının konumu ile geometrisi ve kesici geometrisi dikkate alınarak tanımlanmalıdır.

Bir parçanın işlenmesi için gerekli kesici hareketi, birbirinden ayrı olarak kaba ve ince olmak üzere iki kısımdan meydana gelir. Kaba işlemede kesici hareketinde temel gerekler şöylece sıralanabilir:

1. Mümkün olan ölçüde basit ve tercih edilebilir doğrusal işleme olmalıdır.
2. İşleme zamanını en aza indirmelidir. Diğer bir deyişle, talaş derinliği ile ilerleme miktarı en büyük, kesici yolu en kısa olmalıdır.

İnce işleme hareketinde işlenmesi gereken parça profili Bölüm 9.3.2'de tanımlandığı gibi olmalıdır.

Kesici yolunun tasarlanmasından önce, sistem özellikle o anki bağlama anında hangi yüzeyin işlenebilir olduğunu kontrol etmelidir. Kaba işlemedeki



Şekil 9-8 Yüzeylerinin ve varsayılan sınır düzlemlerinin kaba işlemeyi basitleştirmek için eklenmesi.

kesici hareketi genellikle X, Y ve Z eksenlerinde basit doğrusal hareket olup, çoğunlukla koordinat düzlemleriyle paralel düzlemler kesicinin kaba işlemedeki izlemesi gereken düzlem veya yüzeyleri oluşturmalıdır (Şekil 9-8). Koordinat düzlemleriyle paralel, varsayılan sınır düzlemleri parçanın işlenmeyecek kısımlarına eklenebilir. Kaba işleme için uygun kesici yolu, kaba ölçülerdeki malzemeden kaldırılacak talaş miktarının tesbit edilmesi ile hazırlanır. Talaş miktarı da kaba iş parçasından, işin son halinin sınır düzlemlerinin belirlediği yüzeylerin çıkarılması ile bulunur.

Kaba işlemin esas parçaları, bir cep hareketi olacak biçimde gözönünde bulundurulmalıdır. Örnek: Talaş kaldırma belirli sınırlar içinde katman katman yapılır. Böylece cep hareketi, genellikle zigzag yapacak kesicinin kaba işleme hareketini yönlendirmesi üzerine hazırlanabilir. Örneğin cep işleme hareketi X (veya Y) doğrultusunda Y yönünde (veya X) ilerleme hareketi prensibine sahiptir. Bir pasoluk (katman) talaş kaldırıldıktan sonra, kesici tanımlanan maksimum talaş derinliği kadar ilerler ve cep işleminin ikinci safhasını işleme koyar. NC programı oluşturma sisteminin özü, iş üzerinden kaldırılacak toplam talaş miktarının birkaç cep işleme alanına uygun şekilde bölünmesidir.

Bütün hareketlerin doğrusal özellikli olmasından dolayı, kaba işlemede kesici hareketleri, iş üzerinden kaldırılacak talaş sınırları ile kolayca hazırlanabilir.

İnce işleme, yarı ve tam olmak üzere iki kısma ayrılabilir. Çünkü yarı ince işleme ile, bitirme işlemi için uygun parça yüzeyinin hazırlanmasına imkan verir. Kesici (veya kesiciler) bu işlem süresince parça profilini izler. Kesicinin profili oluşturma hareketleri Bölüm 9-3'de tanımlanan metotla sağlanabilir. Parçanın birçok farklı geometrik özelliklerden meydana gelmesi halinde, sistem hangi özelliklerin bitirilmesi gerektiğini, sınır durumlarını ve ince işlemenin her bir adımı için kesici yolunu hazırlamalıdır. APT ve diğer yüksek düzey NC dillerindeki ana program sonra oluşturulabilir. Sonuçta NC programı, kaynak (ana) programın NC işlemi veya son işlemcinin kullanımı ile hazırlanır.

Yukarıda verilen tanımlarda da görüldüğü gibi, bir CAD modeline dayalı NC programının otomatik oluşumu, büyük ölçüde bir programcı veya işlem planlamasının bilgi taşıma ve karar vermesine dayanan bir işlemdir. Hem kaba hem de ince işleme için, iki merkezi problem tekrar çözülmelidir: (1) İşlenecek alanların tesbiti ve değiştirilmesini, işleme sırasını, kesici yolunu, kullanılacak kesicileri ve işleme özelliklerini içeren tezgah işleme planı, (2) kesici konumlarının hesabı. Kesici konumlarını NC işlemcisinde var olanlara benzer matematiksel yollarla hesaplanmasından sonra, sistemin ana fonksiyonu işleme bilgileri, malzeme, bağlama kalıpları, kesiciler gibi bilgileri içeren üretim bilgi tabanı ve CAD modeli ile hazırlanan bilgi tabanına dayanan tezgah işleme planından ibarettir. Uzman deneyimlerine dayanan yapay zeka programı planlama modülünü oluşturması için sistemle bütünleştirilebilir. Kararlar alınır ve planlama, parçanın CAD modeli (tercihen katı model) ile hazırlanan bilgi tabanına bağlı olarak yapılır. Bundan dolayı, teknik resimleri okuyarak gerekli bilgiyi, elde edilen imalat mühendisi tarafından yapılan işe eşdeğer olan dolaylı CAD modelinden bilginin çıkarılması ve doğru yorumlanan, yapay zeka programlarının çok önemli parçasını teşkil eder.

SORULAR

- 9.1 Temel CAD sistemi (bilgisayar destekli çizim), ve CAD/CAM sistemi arasındaki belli başlı farkları açıklayınız.
- 9.2 CAD/CAM sistemli tasarım ve analiz işleminde mühendisin oynayacağı rol nedir?
- 9.3 Bir parçanın matematiksel veya CAD modeli nedir?
- 9.4 CAD veya CAD/CAM sistemi içinde CAD modeli oluşturmasının anlamı nedir?
- 9.5 CAD sistemleri içinde bir parçanın tanımlanması için kullanılan temel model nedir? Herbirinin üstünlük ve mahzurlarını vererek karşılaştırınız.
- 9.6 Bir CAD/CAM sistemi, bir parçanın tel çerçeve modeline bağlı çevresel hareketini nasıl tanımlar?

- 9.7 Bir CAD/CAM sistemi, tanımlanan bir yüzeye bağlı bir yüzey çevresel hareketini nasıl tanımlar?
- 9.8 Bir CAD/CAM sisteminde ince işlemede kesici yolunun otomatik oluşumu için ihtiyaç duyulan teknik(ler) ve tip(ler) nedir? Hem kaba hem de ince işleme için kesici yolu CAD/CAM sistemi ile otomatik olarak oluşturulacak ise, en uygun modelleme tekniği nedir? Niçin?
- 9.9 Tam bir tezgah işlemi için kesici yolunun otomatik oluşturulmasında çözülmesi gereken temel problemler nelerdir?
- 9.10 Bir CAD modeline dayalı NC programlarının otomatik oluşumunda yapay zeka tekniklerinin oynadığı rol nedir?

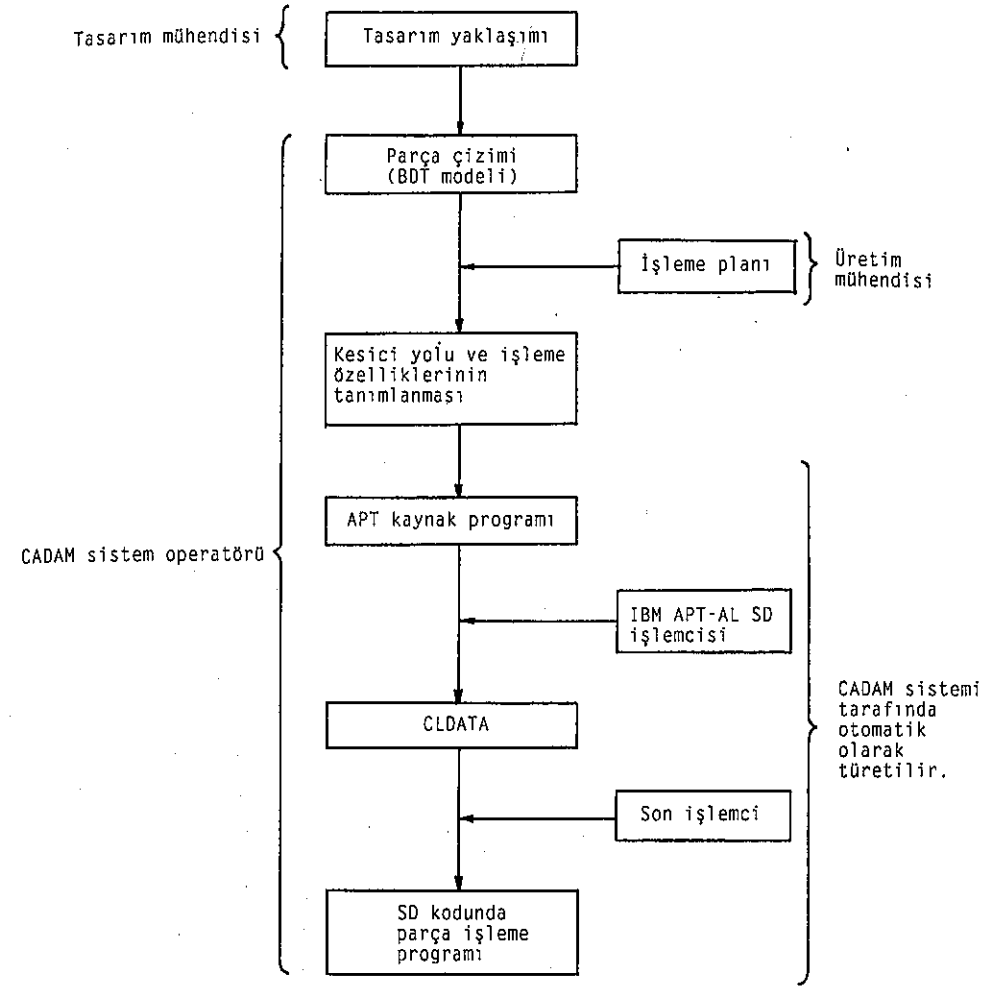
Bölüm 10

CADAM Sistemi ile NC Programlarının Üretimi

Bir örnek olarak, CAD/CAM sisteminde bir CAD modelinden NC programı üretmek üzere CADAM sistemini kullanacağız.

Bir NC programı genellikle işleme özelliklerini ve kesici yolunu tanımlayan durumlardan meydana gelir. Elle ve bilgisayar destekli APT programlamada işin önem verilmesi gereken kısmı, kesici yolunun, tanımlanması ile ilgilidir. İşleme özelliklerinin tanımlanması NC veya APT'de doğru sözcüklerin, komut seçilmesini ve en az hatayı verecek doğru parametrelerin tanımlanmasını içerir. NC programının türetilmesi için bir CAD/CAM sistemi kullanıldığında, durum temel olarak aynıdır. Kesici yolunun tanımlanması hala işin esas kısmını oluşturur. Bununla beraber, programlama cümlelerinin yazımı ve bileşenlerinin özellikleri, belirli bir sırada ekrandan uygun elemanların seçimi ile yer değiştirir, böylece bir programda sözdizimi hataları yapma ihtimali önemli derecede azalır. Bundan başka, tanımlanan geometrik özellikler ekranda görünüp, kontrol edildiği için hatalar daha da azalır. Bir NC programının CADAM sisteminde oluşum sırası Şekil 10-1'de verilmiştir. Bu işlem, programın kullanıcı yardımı ile bir CAD/CAM sistemi ile kurulmasının haricinde APT programındaki ile temelde aynıdır.

Bir CAD/CAM sisteminin CRT ekranında kesici yolunun tanımı, kesici yolunun APT'de tanımlanmasıyla benzerlik taşır. Kesici yolu, parça profili elemanlarının ekranda doğru sıra ile seçimiyle tanımlanır. Sürekli kesici yolu, sürekli profil



Şekil 10-1 CADAM Sisteminde bir NC Programının Türetimi için Akış Şeması

esas alınarak oluşturulur. Bununla beraber ekranda sürekli olarak görülen profil iş sayısal yuvarlamalardan dolayı tam süreklilik göstermeyebilir. Bundan dolayı da NC kesici yolunda problemler meydana gelir. Sonuç olarak CADAM NC fonksiyonunun detaylı bilgisi, kesici yolunun başarı ile tanımlanmasında önemli rol oynar.

Oldukça karmaşık profil ve biçimler için gerekli kesici yolunun oluşmaması durumlarında bu işlem CAD sistem ekranında yapılmalıdır. Bir seri kesici yolu birbirini üzerine bindirilebilir. Bu, istenilen kesici yolunu diğerlerinden ayırmada problemler doğurur. Ayrıca CADAM'daki tel çerçeve modelin kullanımı gerekli

geometrik elemanların tanımlanması için çok fazla sayıda adımı gerektirir. Sonuç olarak (Kısım 8.4'ün 2. örneğinde gösterildiği gibi) üç boyutlu karmaşık profilli parçalar için işlem çok zaman kaybettirici olmaktadır. Bundan dolayı da etkili bir şekilde oluşturulamaz. Böyle durumlarda, CATIA gibi yüzey veya katı modellemeye dayanan CAD/CAM sistemleri veya APT'deki tezgah işlemini programlama bile daha etkili olmaktadır.

Genel olarak bir CAD/CAM sistemi boyunca NC programının oluşumu, çok basit ve kolaydır. Bundan başka kesici yolu ekranda görüntülenebilir ve ekranda adım adım izlenebilir. Böylece bu işlem, programın görsel olarak düzeltilme ve doğrultmasına imkan verir.

Bundan sonraki kısımlar IBM VM/CMS işletim sistemi ve CADAM Release 20-1'e dayanmaktadır.^{13,14}

10.1 CADAM SİSTEMİNDE BİR NC PROGRAMININ ÜRETİMİ İÇİN İŞLEM SIRASI VE KONTROL AKIŞI

Önceden belirtildiği gibi, CADAM sistemi, saklanması için ortak veri tabanı ve geometrisinin çıkarılmasını ve NC bilgisini sağlar. CADAM sistemi içindeki APT arabirim modülü CADAM NC bilgisinin APT-AC NC işlemcisi ile bağlantı kurmasına imkan verir. CADAM çizim dosyaları NC program türetimine ihtiyaç göstermez. Bundan dolayı, APT işlemi süresince bunlara ulaşamaz.

Bir parça ekranda bir kez tanımlandığında, CADAM işleticisi tezgah işlem sırasını ve işlem planına bağlı olarak kesit yolunu tanımlar. NC bilgisi, CADAM NC fonksiyonu altındaki POST alt fonksiyonu yardımıyla saklanır.

CADAM işleticisi, kontrol altındaki ana terminalde çalışır. NC bilgisi APT arabirim modülü ile çağrılır ve sonra APT-AC NC işlemcisi ve seçilen bir son işlemcisi ile işlenir. Sonuçta tezgahla kullanılacak NC programı hazırlanmış olur. CADAM NC bilgisini işlemeden önce, kullanıcı terminal bilgisini, NC bilgisini CADAM sistemine girilmesinde üstünlük sağlamak üzere kesinleştirilmiştir. Bu CADAM sisteminin¹³ yerleştirilmesi esnasında oluşturulan iki dosya ile sağlanır:

REQUEST IDS A1

ve

GROUP OWNER A1

Bu iki dosya, CADAM modülleri ile birlikte kullanıcının bir A diskette saklanmalıdır. Kullanıcı ayrıca son işlemciye ulaşabilmelidir. Bu da aşağıdaki dosyanın kullanıcı diskette saklanması ile sağlanır:

MACHINE STMTS A1

Bu dosya, CADAM grafik terminalinde tanımlanamayan gerekli durumların CADAM ile oluşturulan APT programına uygulamak için kullanılır. Genellikle, MACHIN komutu arzu edilen son işlemcinin seçimi için bu dosyada bulunabilir. Bu dosya normal olarak bir kaç set APT yazılımı içermektedir. Ayrıca her bir set son işlemciye bağlı bir MACHIN komut ilişkisini ve diğer APT yazılımlarını içerir ve belirli bir SD tezgahında kullanılacak CADAM NC bilgisinin işlenmesi için kullanılır. CADAM arabirim modül aşağıdaki dosyadan 73. sütundan 80. sütuna tanımlanmış karakterlerle doğru yazılım gruplarını seçer:

MACHINE STMTS A1

73. sütundan 76. sütuna kadar olan karakterlerle kullanılacak tezgah tanımları ve 77. ile 80. sütunlar seçilecek son işlemciyi belirlemede kullanılır. Her bir tezgah için bir son işlemci kullanılacak ise 77'den 80'e kadar olan sütunla boş bırakılır. Örneğin; bir torna ve freze tezgahı için SD bilgisinin işlenebilmesi için eklenmesi gereken üç veya dört yazılımın olduğunu varsayalım. "MACHINE STMTS A1" dosyası ekranda aşağıdaki gibi görünür:

SÜTUN 1	SÜTUN 73
MACHIN/.....	LATH
.....	LATH
.....	LATH
MACHIN/.....	MILL
.....	MILL
.....	MILL
.....	MILL

73'den 76'ya kadarki sütunlarda tanımlanmış karakterler alfasayısal karakterlerin bir bileşimi olabilir. Bununla beraber, her bir setteki karakterler aynı olmalıdır. Örnek olarak yukarıdaki dosyada LATH ve MILL boş bırakılmıştır. Her bir tezgah için, bu dosya içindeki her set için 20'ye kadar durum tanımlaması yapılabilir. Dosya içinde tanımlanan bilgi setleri için bir sınırlama yoktur.

NC bilgisinin işlenmesinden önce, kullanıcı CADAM programı ve APT-AC NC işlemcisine diskette yardımcıyla kullanacağı tezgahla bağlantı kurmalıdır. Gerekli işlem sırası bu disketteki işlem süresince devreye sokulmasıyla oluşturulmalıdır.

Yukarıda açıklanan tüm işlemler uygulandıktan sonra kullanıcı CADAM NC bilgisini işleme durumuna gelmiştir. Bunun için, yapacağı iş aşağıdaki komutu

yazmaktır:
CADAPT

Bu komut iş oluşum durumunu ve "CADAPT EXEC" içeren CADAM programının ürününü hatıra getirir ve işlemi başlatır. İşlem üç adımdan meydana gelmiştir.¹³ Birinci adımda CADAM sistemi girdi olarak aşağıdakileri kullanır.

1. CADAM veri tabanında saklı NC bilgisi.
2. "MACHINE STMTS A1" dosyasından seçilen bilgi seti.

Sistem "MACHINA STMTS A1" dosyasından alınan yazılımları içeren APT programını ve NC işlemcisinin çeviri bölümündeki GRAPT modülünü meydana getirir bir durum oluşturur. Ayrıca NC işlemcisi tarafından bir birim olarak işleme sokulabilmesi için bir sıra dahilinde NC bilgisini toplar.

İkinci adımda, birinci adımda oluşturulan iki bilgi seti girdi olarak ele alınır. APT programı okunduğunda, GRAPT modülü harekete geçirilir. NC bilgisi okunur ve standart formata dönüştürülür. Dönüştürme safhasından sonra, standart APT işlemi çalıştırılır. Bu adımla NC programı oluşturulur.

Üçüncü adımda, CADAM programı içindeki bir alışlagelmiş çalışma biçimiyle sonraki işlerde kullanılmak üzere bir çok bilgi seti oluşturulur.

10.2 TEZGAH İŞLEMLERİNİN VE KESİCİ YOLUNUN EKRANDA TANIMLANMASI İÇİN YÖNTEMLER VE İŞLEM SIRASI

Yukarıdaki açıklamalarda görüldüğü gibi, bir NC programının türetimi, kesici yolunun ve bilgisayar grafik sisteminin ekranında işleme operasyonlarının tanımlanmasına indirgenir. Programlama için harcanacak gayret, ekranda geometrik elemanların uygun sıra ile ve gerekli kelimelerin seçimi için harcanır.

Böylece CAD/CAM sisteminin NC fonksiyonunun becerili kullanımı daha önemli olur. CADAM içindeki kesici yolunun tanımlanması için işlem sırası APT programlama mantığı çerçevesinde yapılır. Böylelikle APT bilgisine sahip bir operatörün yazılımı anlaması daha kolaylaştırılır. Bu bölümde CADAM ekranında, bir parçanın tezgah işlemlerinin tanımlanması için işlem sırasının detayı verilecektir.

10.2.1 CADAM Grafik Ekran Terminaline Giriş

Bir CADAM grafik terminali veya konsolu dört parçadan meydana gelmiştir. CRT ekranı, fonksiyon klavyesi, alfasayısal klavye ve ışıklı kalem veya imleçli tablo (Şekil 10-2). Ayrıca girdilerin işlenip, ana bilgisayara gönderilip, burada daha

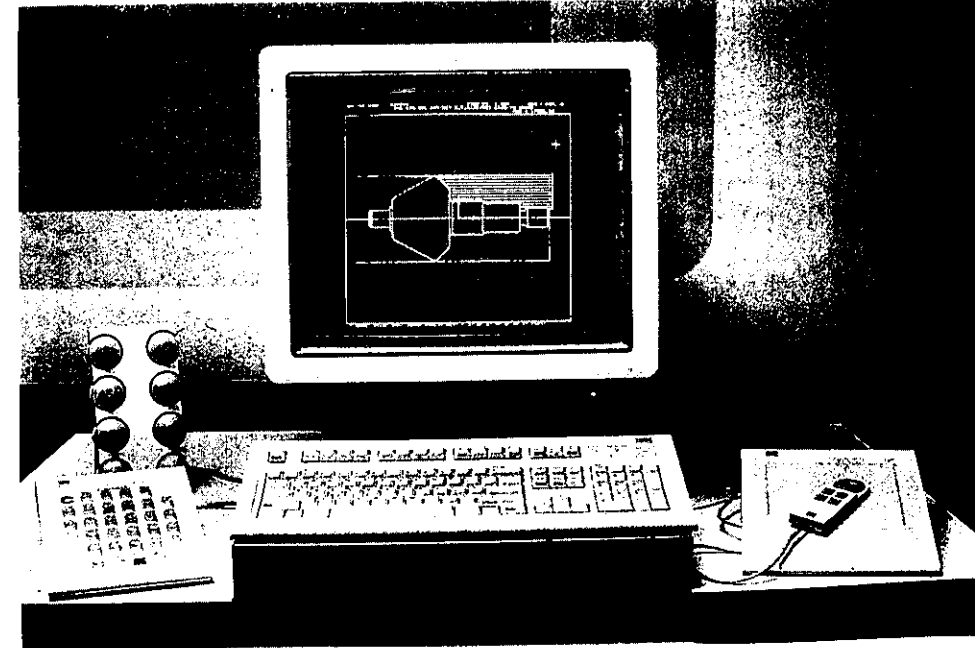
ileri bilgiler için işleyen ve hafızada saklayan denetim bilgisayar çizginin tanımlanması gereken girdileri cevaplamak suretiyle sonuç ekranda görüntülenir. Bir NC fonksiyon tuşuna basıldığı zaman, sistem NC tezgah işlemi için gerekli girdilerin verilmesi için hazır duruma getirilmiş olur.

Alfasayısal klavye, CAD sistemi içinde komutların, fonksiyonların, harf, sayı veya diğer sembollerde meydana gelmiş bilgilerin operatör tarafından girilmesine imkan sağlar. Doğrulama için bir girdi mesajı öncelikli ekranda belirlenir (Şekil 10-4'de 10 ncu nota bakınız) ve bilginin ana hafızaya saklanmasından önce muhtemel düzeltmeler ve doğrulamalar bu ilk mesaja göre yapılır. Klavye yardımıyla ayrıca ekrandaki imlecin yerleşiminin kontrolü de sağlanır.

İşıklı kalem, CRT ekranında ışık demetini tarar. Kalem biçiminde olup, kullanıcıya ekrandaki görüntünün herhangi bir elemanını veya belirli bir konumu işaretleme imkanı verir.

CRT ekranı temel olarak üç alana bölünebilir (Şekil 10-4) Merkezi alan kullanıcı tarafından oluşturulan geometrik elemanlar ve cisimlerin görüntülenmesini sağlar. Ekranın üst kısmı, çizim ölçeği, seçilen fonksiyon, pencere boyutu, hata mesajı ve sistem bilgisi gibi çeşitli mesajların verilmesi amacıyla kullanılır.

Ekranın alt kısmı, seçilen bir fonksiyonun altındaki çeşitli fonksiyonların görüntülenmesini sağlar. Ayrıca kullanıcı tarafından bilgisayar hafızasına verilen



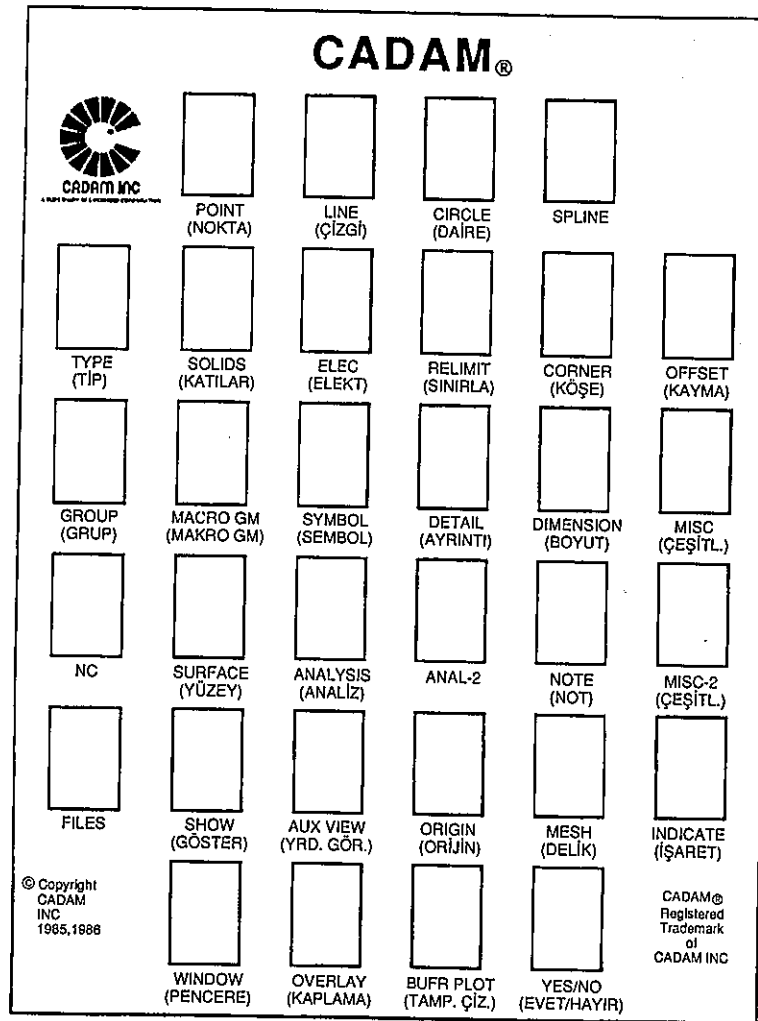
Şekil 10-2 CADAM sistemi iş istasyonu şu parçalardan meydana gelir: CRT ekranı, fonksiyon klavyesi, alfasayısal klavye, imleçli veya ışıklı kalemli tablo (CADAM INC.'den izinli fotoğraf).

girdi görüntülenmeden önce bu alanda, kontrol edilir.

Kontrol birimi, yüksek hızlı bilgisayar ile düşük hızlı görüntü konsolu arasında bir bellek gibi görev üstlenir. Denetim birimi birkaç görüntü konsolunu denetler.

10.2.2 NC Fonksiyonu Altındaki Bir NC'li Tezgah İşleminin Tanımlanması

Bir tezgah işleminin tanımlanmasından önce, bir geometrik çizim veya model, bir-iki veya çok görünüşü verilerek tanımlanmalıdır. Okuyucu bu amaçla CADAM bilgisayar destekli tasarım işleminin detayları hakkındaki bilgileri, CADAM kul-

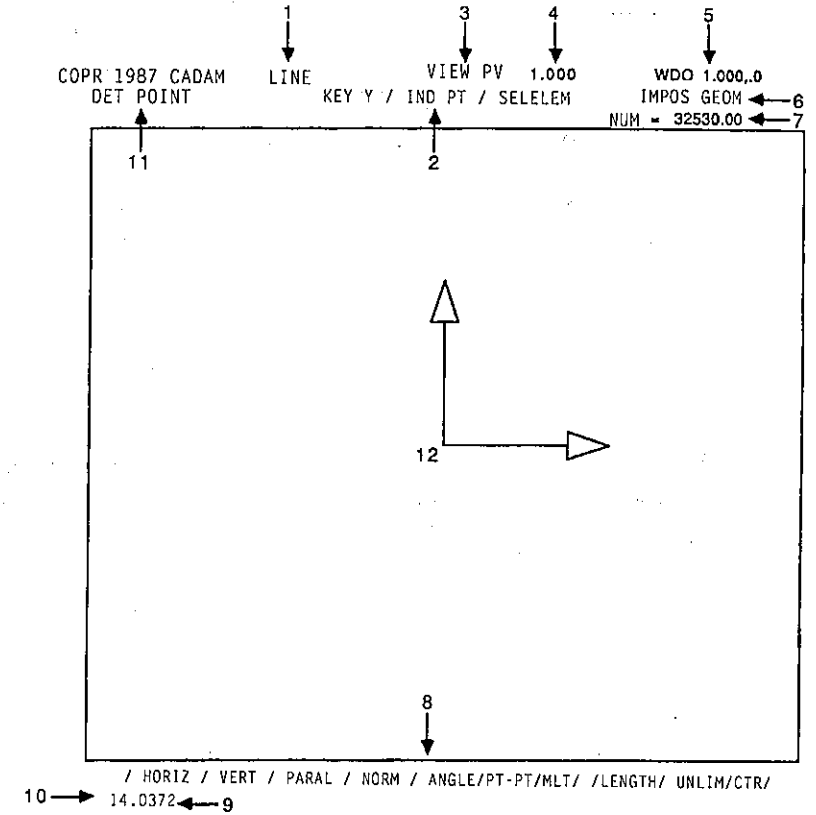


Şekil 10-3 CADAM sisteminin fonksiyon klavye şablonu (Courtesy of CADAM INC.)

lanım kılavuzundan¹⁶ temin etmelidir. NC'li tezgah işlemi ise CADAM sisteminin NC fonksiyonu altında tanımlanmış ve bununla ilgili tanımlamalar aşağıya çıkarılmıştır.

Fonksiyon klavyesinde bir NC fonksiyonu tuşuna basıldığında, ekranın alt kısmında aşağıdaki gibi bir alt fonksiyon listesi tanımlanır:

/ENTER/PASS/ERASE NC/PACK/NEST/PLOT/EXIT/



Şekil 10-4 CADAM sisteminin grafik görüntü ekranı. Tuşlar:

1. Seçilen fonksiyon
 2. Bilgisayar mesajı (kullanıcıya o anki işleme adımını bitirme için seçme imkanı veren seçenekler listesi)
 3. Görüntü tanımlayıcı
 4. Ölçek
 5. Pencere parametreleri
 6. Hata mesajı
 7. Sayaç
 8. Menü (seçilen fonksiyon altında çeşitli alt fonksiyonlar listesi)
 9. Tahribatsız imleç
 10. Geçici asma kaydedici
 11. Bilgi
 12. Orijin belirteci
- (Courtesy of CADAM INC.)

Bu fonksiyonların açıklaması Şekil 10-5'de verilmiştir. Bir alt fonksiyon ışıklı kalemle kullanımı ile bu listeden seçilebilir. PASS fonksiyonu genellikle tezgah işleminin başlatılması amacıyla kullanılmaktadır. PASS fonksiyonu seçildiğinde, ekranın alt kısmında aşağıdaki menü belirir:

/START/REPLAY/RESTART/POST/ABORT/LEAVE/

Bu fonksiyonlarla ilgili açıklamalar Şekil 10-6'da verilmiştir. Bu adımda tasarım ihtiyaçlarına göre bir fonksiyon seçilebilir. Örneğin yeni bir tezgah işleminin tanımlanması gerekiyorsa START fonksiyonu seçilmelidir. Şayet ara verilen bir tezgah işlemi var ise buna devam için RESTART fonksiyonu seçilmelidir. REPLAY fonksiyonu, kullanıcıya YES/NO (EVET/HAYIR) tuşlarına basarak ekranda kesici hareketlerini ve komutlarını adım adım okuma imkanı verir. Eğer tezgah işlemi doğru bir şekilde tanımlanmış ise, bilgi POST fonksiyonu altındaki bir veri tabanında saklanır. ABORT fonksiyonu ise önceden tanımlanmış NC bilgisini silme imkanı vermektedir. LEAVE fonksiyonu ise Şekil 10-5'de gösterilen menüye dönmeyi temin eder.

10.2.2.1 Bir Tezgah İşleminin Tanımlanmasına Başlamak için Kullanılan İşlem Sırası

START (BAŞLA) fonksiyonu bir tezgah işlemine başlayabilmek için seçilmelidir. Bu seçim APT programlamadaki ilk durum olan PARTNO'nun tanımlanması ile benzerlik taşır. Bu fonksiyon seçildiğinde sistem aşağıdaki mesajı kullanıcıya yöneltir (Şekil 10-7).

SEL PASS TYPE/KEY CUTTER DIA = ()

Kullanıcı öncelikle NC'li tezgah işlemini seçmeli ve sonra kesici çapını belirtmelidir. Bu durumda, ekranda beliren görüntü, daha önce seçilmiş olan ve 3-EKSEN'i tanımlayan aşağıdaki menüdür:

/3-AXIS/MULTAX/

3-AXIS (3-EKSEN) seçeneği işlemin üç eksenli bir tezgaha gerçekleştirileceğini ifade eder (tornalama gibi; iki-eksenli işleme; bu seçenek altında tanımlanabilir). İşleme anında kesici ekseninin yönündeki bir değişmeyi içeren bir tezgah işlemi MULTAX (ÇOKEKSEN) seçeneği altında tanımlanabilir. Burada 3-AXIS durumu incelenecektir. Eğer tezgah işlemi 3-AXIS çeşidi ise, kullanıcı ilk seçeneği, SEL PASS TYPE (3-AXIS öncelikli seçildiğinden) geçip, alfa sayısal klavyeden kesici

ENTER / PASS / ERASE NC / PACK / NEST / PLOT / EXIT

SD tipinin geometrik elamanları bu fonksiyon altında yapılandırılmıştır.

Bu fonksiyon altında, kullanıcı 2-D (2-B), 3-D ve çok eksenli kesici yolunu oluşturabilir, daha önce oluşturulmuş kesici yoluna SD bilgisini yükleyebilir, var olan kesici yolunda değişiklik yapabilir ve SD işlemcisi ve son işlemci ile işlenmesi için oluşturulan SD bilgisini saklayabilir.

Bu fonksiyonun seçimi ile bu bölümde oluşturulan tüm SD bilgisi silinebilir.

Saklanmış bilgi, hafıza boşluklarından faydalanabilmesi için daha uygun bir formda düzenlenebilir.

Bu fonksiyon, kullanıcıya GROUP fonksiyonu altındaki uygunluklardan faydalanma imkanı verir.

Bu fonksiyonun seçimi, ekrandaki görüntünün bir çizici yardımıyla çıktısının alınmasına imkan verir.

SD fonksiyonunda çıkış

Şekil 10-5 NC alt fonksiyonları için bir açıklama.

cinin çap değerini (tornalama kesicisi için kesici uç yarıçapıdır) girer. Çap girdi olarak kabul edildiğinde, sistem kullanıcıya aşağıdaki mesajı verir:

KEY COR-R = ()

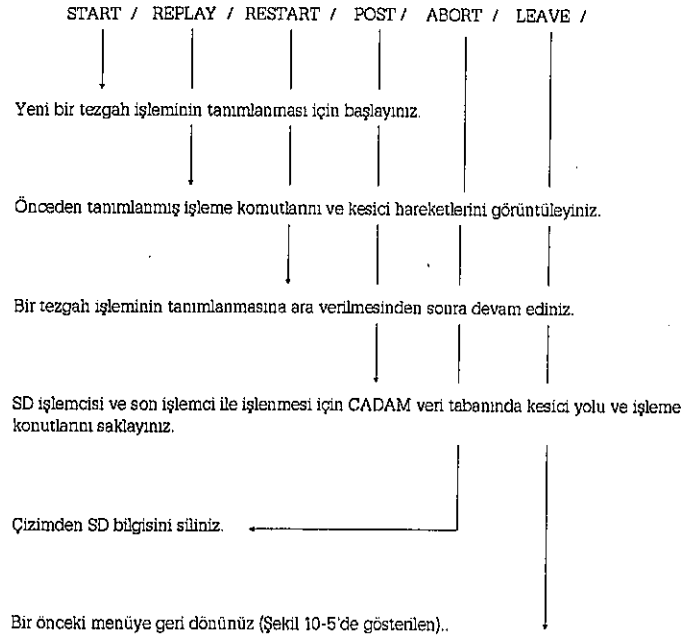
Bu mesaj ile kullanıcıdan parmak frezenin köşe yarıçap değeri (tornalama kesicileri ve düz-parmak frezeler için bu sıfır değeridir) istenir.

Yukarıda açıklanan iki adım APT programlama içinde CUTTER (KESİCİ) tanımlanmasındaki aynı etkilere sahiptir.

Yukarıdaki menüden sonra sistem aşağıdaki mesajı ekrana getirir:

Y/N NEXT AXIS/SEL LOAD PT

İlk seçenek, Y/N NEXT AXIS (sonraki eksen dönüşümünü seçmek için YES/NO



Şekil 10-6. PASS fonksiyonu altındaki alt fonksiyonlar.

fonksiyon tuşunun kullanımı) CLDATA (kesici yolu bilgisi)'nin dönüşümünü tanımlamak için kullanılır. Eğer herhangi bir dönüşüme gerek yok ise, SEL LOAD PT (başlangıç noktasının seçimi) işlemi harekete geçirilir. Burada kullanıcı başlangıç noktası olarak X-Y düzleminde daha önceden tanımlanmış bir noktayı ışıklı kalem ile seçebilir. Bu işlem APT programlamanın başlangıcındaki FROM deyimini ile tanımlanan başlangıç noktasının tanımlanması ile benzerlik taşımaktadır.

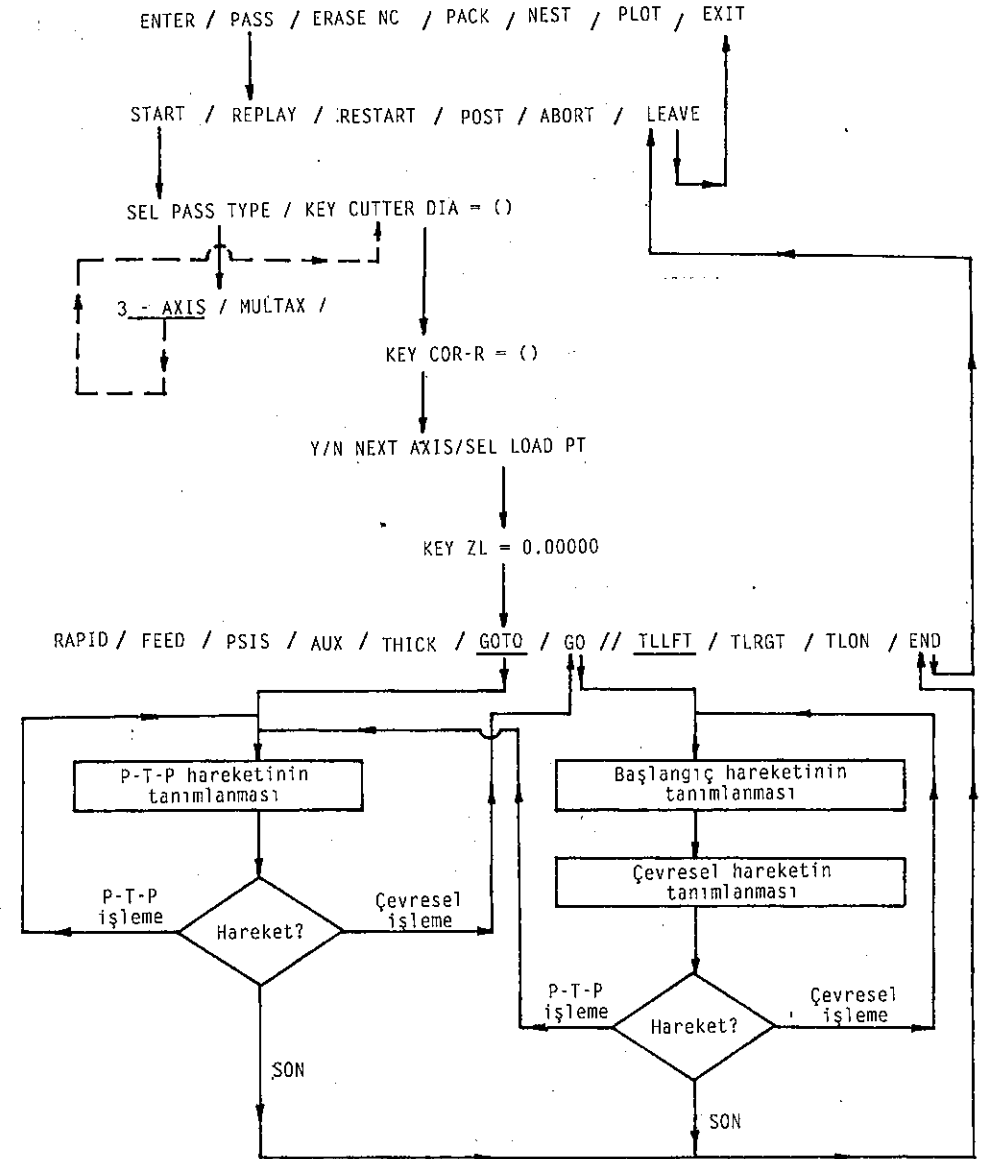
Sistem daha sonra kesicinin Z eksenindeki ilk konumunu aşağıdaki mesaj yardımıyla sorar:

KEY ZL = 0.00000

Bu örnekte, kesicinin Z doğrultusundaki ilk konumu sıfır olup, kullanıcı, bunu istediği herhangi bir değer ile klavyeden girmek suretiyle değiştirebilir.

10.2.2 Noktadan Noktaya ve Çevresel İşleme Hareketinin Tanımlanması için Kullanılan İşlem Sırası

Yukarıda belirtilen öncelikli işlemlerin tanımlanmasında, kesici hareketini ve tezgah işlemlerini tanımlayan menü ekranda aşağıdaki gibi oluşur:



Şekil 10-7. Bir tezgah işleminin tanımlanması için işlem sırası. Altı çizili fonksiyonlar sistem tarafından menü ekranda görüntülediğinde çıkmaktadır. P-T-P: Point To Point (Noktadan Noktaya).

/RAPIDO/FEED/PSIS/AUX/THICK/GOTO/GO/TLLFT/TLRGT/TLON/END/

APT'deki aynı anlamlarla kullanılan GOTO ve TLLFT fonksiyonları öncelikle seçilir. Aslına bakılırsa TLLFT seçeneğinin öncelikli seçilmesine rağmen GOTO seçeneği altında tanımlanan noktadan noktaya harekette herhangi bir etkisi yoktur. Bu menüde kullanılan kelimeler, APT fonksiyonlarının (APT ve son işlemci komutları) seçimini sağlayan AUX kelimesinin dışında APT'dekininki aynıdır. Kullanıcının seçimi için uygun APT fonksiyonları menüsü ekranda görüntülenir (Şekil 10-8). Kullanıcı CLPRNT veya END gibi APT fonksiyonlarının tanımlanması, devir sayıları veya soğutma sıvısı gibi tezgah özelliklerinin belirlenmesi, ışıklı kalem veya klavyeden uygun tuşları kullanarak APT fonksiyon menülerindeki uygun APT deyimlerinin seçimi ile yapılabilir.

Tezgah özelliklerinin doğru bir şekilde tanımlanmasından sonra, kullanıcı kesici hareketini GOTO modunda (noktadan noktaya hareket) veya GO modunda

COPR 1985 CADAM NUM CON VIEW PV SCL 1.000 WDO 1.000.0
SEL MAJOR WORD

CLPRNT		ATANGL	MANUAL	XYPLAN
GOHOME	DELAY/	AUTO	MAXIPM	YZPLAN
OPSTOP	HEAD/	BORE	MAXRPM	ZXPLAN
PPRINT	INSERT	CCLW	MIST	
RAPID	LEADER/	CLW	NEXT	
STOP	LINTOL/	CSINK	NIXIE	
GTOLER/	LOADTL/	DECR	NOW	
MIRX	MCHTOL/	DEEP	OFF	
MIRY	OPSKIP/	DOWN	ON	+
MIROFF	PITCH/	DRILL	RADIUS	
	PREFUN/	FACE	RANGE	
	REWIND/	FLOOD	REAM	
	ROTABL/	HIGH	RIGHT	
ROTATE/	ROTHED/	INCR	ROTREF	
	SECTL/	IPM	RPM	
	SEQNO/	IPR	SFM	
	SPINDL/	LARGE	SMALL	
AUXFUN/	THREAD/	LEFT	TAP	
COOLNT/	TMARK/	LENGTH	THRU	
CUTCOM/	TRANSZ/	LOCK	TURN	
CYCLE/	TURRET/	LOW	UP	

/POCKET/INDEX/COPY/RETURN/

Şekil 10-8 SD fonksiyonundaki AUX alt menüsünden seçilebilecek APT yardımcı fonksiyonları (Courtesy of CADAM INC).

(Çevresel Hareket) tanımlanmasına başlayabilir.

Eğer Noktadan-Noktaya Hareket, tanımlanacaksa, GO TO deyimini seçilmelidir. Bunun sonucu sistem aşağıdaki mesajı verir.

KEY X,Y/IND PT/SEC ELEM

Bu mesaj ile GOTO modunda, hareket, hedef noktasının X ve Y koordinatlarının verilmesi ile tanımlanabilir. Tanımlama işlemi ışıklı kalemin kullanımı veya önceden tanımlanmış nokta, çizgi veya eğri gibi elemanların seçimi ile yapılabilir. Noktadan noktaya hareketin kılavuz elemanı gibi seçilen eğri veya doğru, eğrinin başladığı hareket üzerinde önceden tanımlı bir noktaya sahip olmalıdır.

Eğer sonraki çevresel bir hareket ise, GO deyimini hedeflenen profili verebilecek kesici konumuna bağlı, uygun seçenikle birlikte seçilmelidir. Örneğin, eğer kesicinin profilin sağ tarafında olması istenirse, TLRGT. (kesici sağda) seçilmelidir. İkinci adım başlangıç hareketinin tanımlanmasıdır.

APT programlamada olduğu gibi, CADAM sisteminde başlangıç hareketinin tanımlanması için bir çok yol vardır. İşin çizimini birkaç görünüşten meydana gelmiş ise başlangıç hareketi ön görünüşe (X-Y düzlemi) veya yan görünüşe göre (Y-Z veya X-Z düzlemi) başlatılabilir. GO fonksiyonu ve uygun tanımlayıcı, TLON, TLLFT veya TLRGT, seçildiği zaman, sistem aşağıdaki mesajı verir;

SEL ELEM:

Bu mesajla kullanıcıdan ilk çevresel harekette kesicinin ön veya yan görünüşte kullanacağı sürücü yüzeyi istenir. Aşağıdaki tanımlama, öngörünüştteki başlangıç hareketinin ifadesine uygulanır (Şekil 10-9 [a,b ve c]).

Bir elemanın seçilmesinden sonra (Şekil 10-9 [b ve c]'deki S1) S1 yüzeyine kesicinin mesafesinin ne olacağı hakkında mesaj, ekranda belirir.

KEY THICK =

Kullanıcı boşluk (mesafe) değerini yazarak cevaplar. Sonra ekranın alt kısmından PSIS fonksiyonunu seçerek parça yüzeyini tanımlar. Ve başlangıç hareketini tanımlamaya devam eder veya başlangıç hareketinin tanımlanmasının tamamlanmasından sonra belirlemek üzere sonraya bırakır. Her iki durumda da ekranda belirlenen mesaj Şekil 10-9 (b ve c)'de gösterilmiştir.

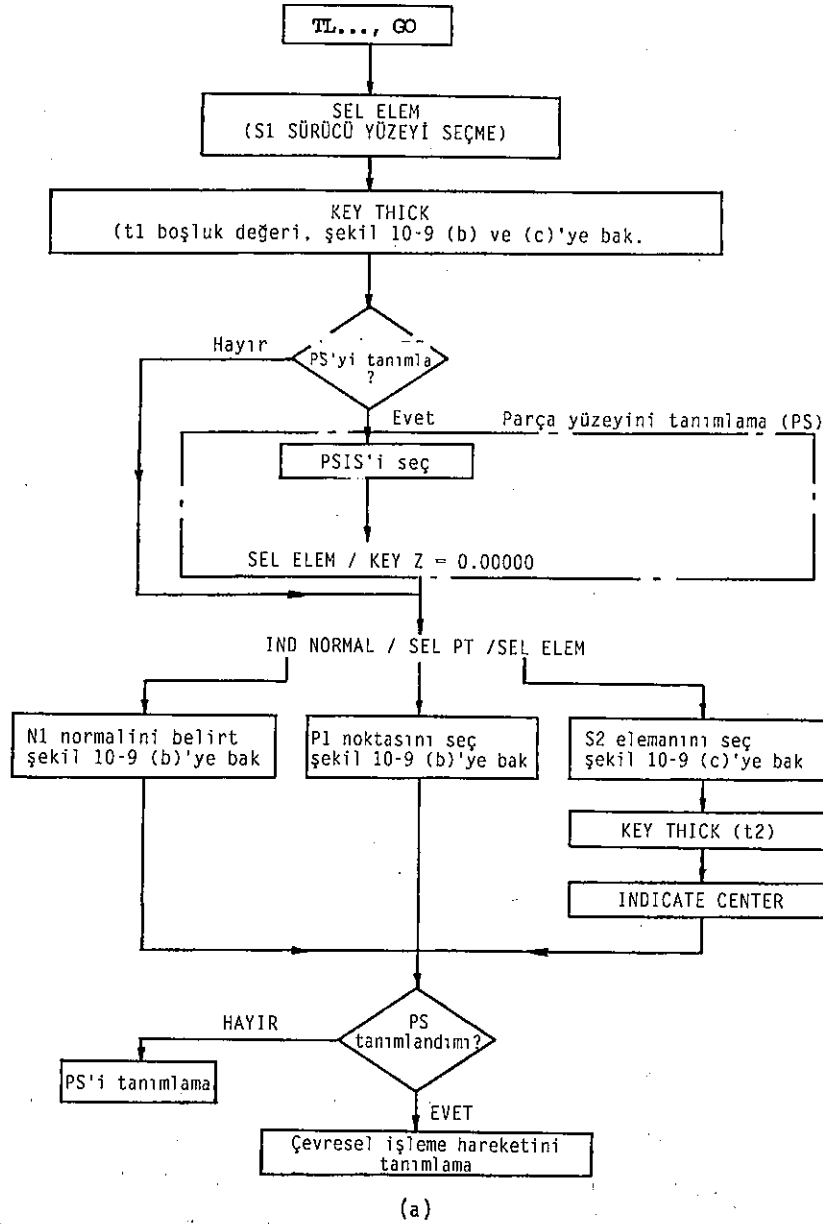
IND NORMAL/SEL PT/SEL ELEM

Bu mesaj ile başlangıç hareketinin bitiş konumu tanımlanır. İlk seçenek ele alındığında (IND NORMAL) (Şekil 10-9 [b] kullanıcı, seçilen S1 sürücü yüzeyinin uygun yerinden geçen N1 normal boyunca bir konumu ışıklı kalemle işaret eder. Bu normal çizgisi başlangıç hareketinin bitiş konumunu hazırlamak amacıyla kontrol yüzeyi gibi kullanılır.

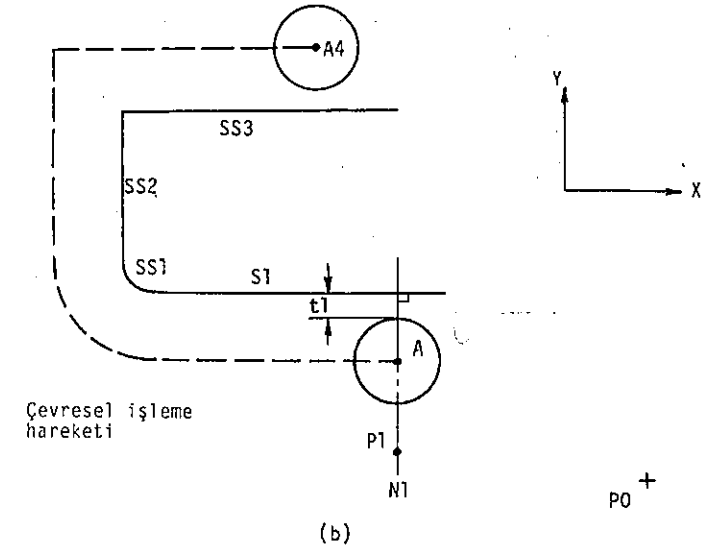
Eğer ikinci seçenek, SEL PT, seçilirse (Şekil 10-9 [b] önceden tanımlanmış

bir nokta (P1) seçilmelidir. P1 noktasından geçirilen N1 normalini, kesicinin son konumunu belirlemek için kullanılır.

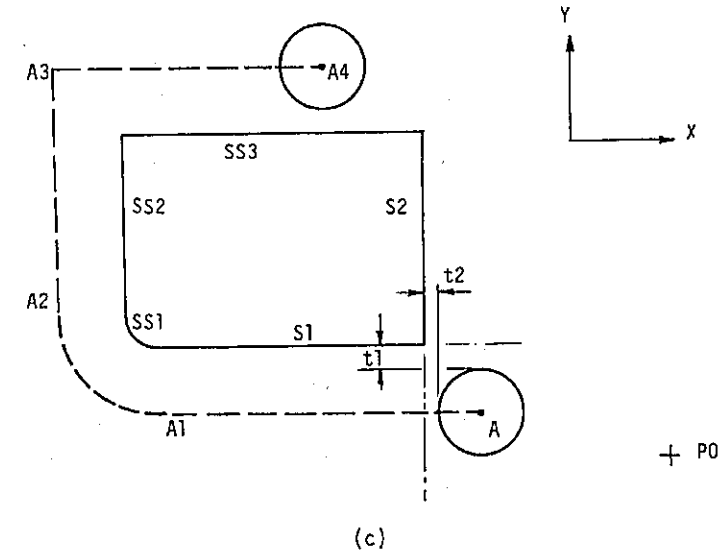
Üçüncü seçenek, SEL ELEM kesicinin seçilen kontrol yüzeyine (S2) bağlı son konumunun belirlenmesi için kullanıcıya imkan verir (Şekil 10-9 [c]). İkinci elemanın seçiminden sonra, (S2) yüzeyindeki t2 mesafesi "KEY THICK" mesajına



Şekil 10-9 (a) Üstten görünüşte başlangıç hareketinin tanımlanması için kullanılan işlemler.



Şekil 10-9 (b) Başlangıç hareketinin son konumunun, A, IND NORMAL seçeneği (N1 normalinin işaretlenmesi) veya SEL PT (P1 noktasını seçimi) ile tanımı. PO: Kesicinin başlangıç konumu.



Şekil 10-9 (c) Başlangıç hareketinin son konumunun A, SEL ELEM seçeneği (kontrol yüzeyi gibi kullanılan S2 elemanının seçimi) t2 boşluğunun kullanımı ve kesicinin konumunun işaretlenmesi ve tanımlanması.

cevap olarak verilmelidir. Eğer TLLFT veya TLRGT seçenekleri seçilir; kesici S1 ve S2 doğrularının belirlediği dört çeyrekte herhangi birinde konumlandırılırsa,

uygun konum ışıklı kalem veya INDICATE fonksiyon tuşunun kullanımı ile belirlenmelidir. Sonra kesici Şekil 10-9 (c)'de gösterildiği gibi uygun bitiş noktasında konumlandırılır.

Başlangıç hareketi uygun şekilde tanımlandıktan sonra, parça profilini meydana getiren yüzeyler aşağıdaki işlemler ile seçilir. Örneğin (Şekil 10-9 [c]). Eğer istenilen profil S1, SS1, SS2 ve SS3 yüzeylerinden meydana gelmiş, kesicinin son konumu A4 ise, SS1, SS2 ve SS3 yüzeyleri seçilebilir (Başlangıç hareketinde tanımlanmasından dolayı S1 yüzeyinin tekrar tanımlanmasına ihtiyaç yoktur). Seçilen her bir yüzey, bir önceki adım için kontrol yüzeyi ve bir sonraki hareket adımı için ise sürücü yüzeyi olarak kullanılır.

Çevresel hareketi oluşturmak için GOTO fonksiyonunun ve profil üzerinde uygun bitiş konumunun (bu örnekte A4) seçilmesi yeterlidir.

Çevresel hareketin tanımlanması süresince, parça yüzeyinin ve tezgah işlemlerinin değiştirilmesi için GOTO seçeneklerinden birinin seçilmesi gerekir (RAPIT, FEED, PSIS, AUX ve THICK).

Bütün tezgah işlemlerinin tanımlanmasından sonra, END seçeneği seçilir (Şekil 10-7). Bu deyim APT'deki END gibi tezgah işleminin sonucunu ifade eder. Sonra yapılan tezgah işleminin adım adım izlenebilmesi için REPLAY seçeneği seçilir ve YES/NO tuşlarına fonksiyon klavyesinde basılarak işlem adım adım takip edilir. İşlemin hem ileri hemde geriye doğru çalıştırılması mümkündür. Bu işlem süresince kullanıcı tezgahın özelliklerinden birini veya kesici yolunun bir kısmını değiştirebilir veya silebilir. Tasarlanmış tezgah işlemi doğrulandıktan sonra POST seçeneği seçilir. Sistemin girdisi olarak iki mesaj kullanılır: (1) APT programlamadaki PARTNO deyiminde tanımlanan yazılıma benzer program tanımlaması aşağıdaki mesajla belirlenir.

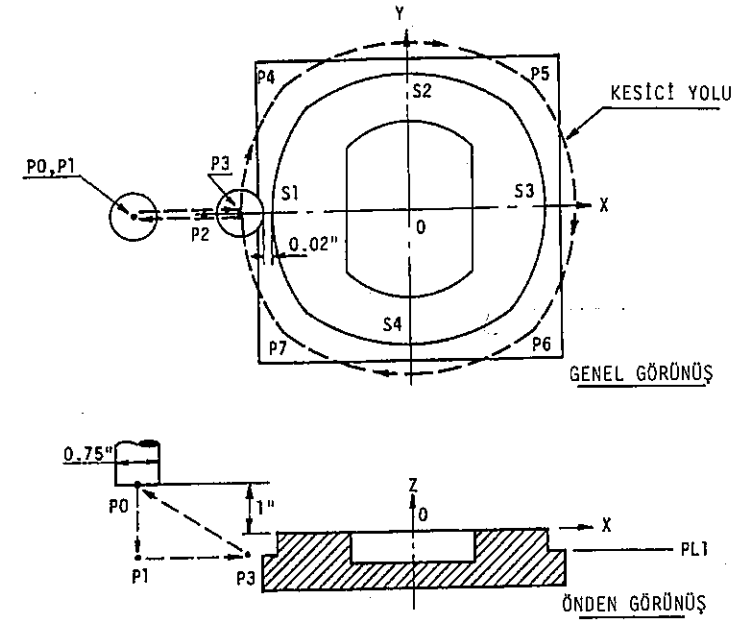
KEY PN, FN / Y/N CURRENT NO

Mesajdaki PN (Part Number) parça numarasının FN (File Number) dosya numarasındaki kısaltılışıdır. İkinci seçenek, o anki dosya ismini YES/NO fonksiyon tuşlarına basarak PN ve FN olarak kabul eder. (2) Tezgah kodu "MACHINE STMTS A1", dosyasındaki 73. ve 76. kolonlarındaki durumlara uygun olmalıdır. Bunun için bilgisayar ekranında aşağıdaki mesaj belirir.

KEY MACHINE =

Yukarıdaki adımların tanımlanmasından sonra, ekranın sol üst köşesinde NC bilgisinin CADAM bilgi tabanında saklandığını gösteren DONE mesajı görünür.

Şekil 10-10. bir NC'li tezgah işlemini tanımlamak için CADAM sisteminin kullanımı örneğini sunar.



Şekil 10-10

CADAM sisteminde NC'li Tezgah İşleminin tanımlanmasında kullanılan işlem sırası

1. Fonksiyon klavyesinden SD fonksiyon tuşuna basınız.
2. PASS fonksiyonunu seçiniz.
3. START fonksiyonunu seçiniz.
4. Kesici çapının değerini giriniz: 0,75"
5. Köşe yarıçap değerini giriniz: 0
6. Üst görünüşten PQ noktasını seçiniz (PQ: başlangıç noktası)
7. Z koordinatını giriniz: 1"
8. AUX fonksiyonunu seçiniz ve devir sayısını, dönme yönünü, soğutma sıvısını vs. belirleyiniz.
9. RAPID fonksiyonunu seçiniz.
10. Önce ön görünüşte P1'i seçiniz. Sonra üst görünüşte P1'i belirleyiniz (GOTO altfonksiyonunun daha önce seçildiğini hatırlayınız). Şimdi kesici P1 noktasına gelir.
11. FEED'i seçiniz ve tasarlanan değeri giriniz.
12. GO ve TLLFT alt fonksiyonlarını seçiniz.
13. Başlangıç hareketinin tanımlanmasına başlamak için S1 yüzeyini seçiniz.
14. Boşluk değerini giriniz: 0,02"
15. PSIS fonksiyonunu seçiniz.
16. PL1 yüzeyini seçiniz (Parça Yüzeyi)
17. P2 noktasını seçiniz (Kesici şu anda P3 noktasındadır)
18. S2, S3, S4 ve S1'i seçiniz.
19. Çevresel hareketi oluşturmak için GOTO fonksiyonunu seçiniz.
20. P3 noktasını seçiniz (Kesici şu anda P3 noktasındadır).
21. Önce ön görünüşte, sonra üst görünüşte PQ'ı seçiniz (Kesici şimdi P3 noktasındadır.)
22. AUX fonksiyonunu seçiniz ve fener milini durdurup, soğutma sıvısını kapatınız.
23. END fonksiyonunu seçiniz.
24. LEAVE fonksiyonunu seçiniz.
25. EXIT fonksiyonunu seçiniz

Bir APT Kaynak Geometri Türetici Programı (ASGG-APT Source Geometry Generator) CADAM sistemi içinde oluşturulmuş tasarım modeline göre APT geometrik tanımlama deyimlerinin NC programcısı tarafından otomatik olarak türetilmesine yardımcı olur. Bu program, programcının karmaşık bir parça için geometrik elemanların tanımlanması ve işaretlenmesi için zaman harcanmasını engeller.

CADAM sisteminin üstünlüklerinden biri, doğrusal enterpolasyona dayanan bir seri GOTO deyimini ile tanımlanan noktadan noktaya hareket serileri şeklinde belirlenen dairesel kesici hareketidir. Böylelikle oldukça çok sayıda doğrusal hareket durumu dairesel hareket için NC çıktısında oluşturulabilir.

10.3 BİR NC PROGRAMININ BİLGİSAYARDAN NC DENETLEYİCİSİNE GÖNDERİLMESİ

APT-AC NC işlemcisi veya CAD/CAM sistemi ile oluşturulan bir NC programı bilgisayar hafızasında saklanır. NC'li tezgah işlemlerinin denetlenebilmesi için bu programın NC denetleyicisine girilmesi gerekir. NC programının bilgisayardan NC denetleyicisine gönderilebilmesi için birkaç yol vardır.

1. Elle bilgi girişi
2. Kağıt şerit veya manyetik kaset hazırlama ünitesi. Ünite programı kağıt şeride delerek işlemek veya manyetik kasete kaydetmek suretiyle çalışır. Şerit veya kaset daha sonra NC denetleyicisine yüklenir.
3. Programı doğrudan bilgisayardan tezgaha gönderme (Bilgisayar, NC deyimlerini işleme süresince teker teker tezgaha göndermek için kullanılır. Doğrudan Sayısal Kontrol olarak isimlendirilen sistemle bilgisayar NC işlemini doğrudan kontrol eder. Veya NC programın tümü NC denetleyicisinin hafıza birimine gönderilir. Bu birime dağıtılmış sayısal denetim, kısaca DNC denir).

Bu üç metodun hepsi üretimde kullanılmaktadır. Bununla beraber, herhangi bir metodun seçimi teknik çevreye bağlı olmaktadır (imkanlar, kullanıcının teknik uzmanlığı ve ürün). Genel olarak, elle bilgi girişi en yavaş ve hata yapma ihtimali yüksek bir işlemdir. Bunun için, bu metod yerine diğer ikisi tercih edilmelidir. Üçüncü metod bilginin aktarımında en kestirme olandır. Doğrudan sayısal denetimde, bilgisayar ve denetim ünitesi imalat süresince etkileşim halindedir. Bilgisayar NC kontrolcüsünün ihtiyaç duyduğu NC yazılımını gönderir. Bundan dolayı tezgah bilgisayarın doğrudan kontrolü altındadır. Genellikle, APT veya CAD/CAM sistemlerini çalıştıran bilgisayar, bir firmanın bir çok bölümü tarafından değişik amaçlar için kullanılabilir. Bu bilgisayar sistemindeki iş yükü, tezgahın

gerçek zamanda kontrolünü gerçekleyemeyecek kadar ağır olabilir. Ayrıca merkezi bir bilgisayar sistemi ile birçok NC'li tezgahın kontrolü, sistemin fonksiyonlarının uygunsuzluğundan dolayı uygulamada fasıllara sebebiyet verebilir.

Modern CNC (Bilgisayarlı Sayısal Denetim)'li tezgahlarda, NC programların saklanması için CNC işlemcisinde hafıza ünitesi mevcuttur. Tamamlanmış NC programı bilgisayardan doğrudan CNC (dağıtılmış sayısal denetim) denetleyicisine yüklenir. Böylelikle bilgisayarın doğrudan sayısal denetimde kullanımı yerine, program NC'li tezgah işleminin denetiminde kullanılır.

CNC denetleyicisi hafızasının yetersiz kaldığı uzun programların çalıştırılmasında en kullanışlı yol, bilginin (NC programının) manyetik kaset veya delikli kağıt şeride yüklenmesidir. Manyetik kaset bilgisayara bu amaçla bağlanmış bir ünite, kağıt şerit ise delikli şeridi hazırlamada kullanılacak ünitenin bilgisayara bağlanması ile oluşturulur.

Son yıllarda, NC programlarının oluşturulmasında ve NC'li tezgahların denetlenmesinde genişletilmiş imkanlara sahip kişisel bilgisayarların (PC) kullanımı hızla artmıştır. Kişisel bilgisayar, daha az karmaşık NC yazılımların kullanımı ile NC programların oluşturulmasında tek başına kullanılabilir. Bunun yanında, kullanıcı tarafından oluşturulmuş programın saklanmasında merkezi bilgisayarın yardımcı şeklinde de kullanılabilir. Her bir durumda, NC programı PC'den CNC denetleyicisine gönderilebilir. CNC denetleyicisi ve PC bilgi naklinde kullanılan RS232 arabirim elemanı ile birbirine bağlanır. EIA RS-232-C standardına¹⁸ göre, iki birim arasında memnun edici iletişim sağlanabilmesi amacıyla birkaç parametre hem PC hem de BSD denetleyicisinde uygun şekilde yerleştirilmelidir. PC tarafında bilginin gönderilme hızı ("baud" değeri *), parity kontrol tipi EIA kodu için ODD ve ISO için EVEN) ve duruş bit (stop bit) sayısı doğru olarak tanımlanmalıdır. CNC denetleyicisi tarafında ise, "baud" değerine (PC'dekinin aynısı olmalıdır) ve kod tipi (EIA veya ISO) parametreleri ve girdi aygıtının uygun şekilde verilmesine ihtiyaç vardır. IBM PC ve FANUC GMB CNC denetleyicisinde bu değerlerin yüklenmesine ait işlem sırası 19 ve 20nci referanslarda tanımlanmıştır.

III. KISIM İÇİN REFERANSLAR

1. Kaplinsky, R., *Computer-Aided Design: Electronics, Comparative Advantage and Development*. New York: Macmillan, 1982.
2. Scrivener, S. A., R., ed., *Computer Aided Design and Manufacture: State of the Art Report*. Oxford, UK: Pergamon Infotech, 1985.
3. Besant, C.B. and Lui, C.W.K., *Computer-Aided Design and Manufacture*, 3rd ed. Chichester, U.K.: Ellis Horwood, 1986.
4. Mortenson, M.E., *Geometric Modeling*. New York: Wiley, 1985.
5. CAD/CAM, *Productivity Equipment Series*, 2nd ed. Dearborn, Mich.: Society

* "Baud" deyimini saniyedeki bit sayısıdır.

- of Manufacturing Engineers, 1985.
6. *Modern Machine Shop 1986 NC/CAM Guidebook*. Cincinnati, Ohio: Gardner Publications, 1986.
 7. Faux, J.D. and Pratt, M.J., *Computational Geometry for Design and Manufacture*. Chichester, U.K.: Ellis Horwood, 1979.
 8. Hyodo, Y., "HAPT-3D: A Programming System for Numerical Control". In *Computer Languages for Numerical Control*, J.Hatvany, ed. Amsterdam: North-Holland, 1973.
 9. Parkinson, A., "An Automatic NC Data Generation Facility for the BUILD Solid Modeling System" In: *16th CIRP Intl. Seminar on Mfg. Systems*, Tokyo, 1984.
 10. Parkinson, A., "The Use of Solid Models in BUILD as a Database for NC Machining". In: *Software for Discrete Manufacturing*, J.P. Crestin and J.F. McWaters, eds. Amsterdam. North-Holland, 1986.
 11. Ferstenberg, R. et al., "Automatic Generation of Optimized 3-Axis NC Programs Using Boundary Files". *Proceedings, 1986 IEEE International Conference on Robotics and Automation*, The Institute of Electrical and Electronics Engineers, 1986, p. 325.
 12. Preiss, K., et al., "Automated Part Programming for CNC Milling by Artificial Intelligence Techniques," *Journal of Manufacturing Systems*, vol. 4, No. 1, 1985, p. 51.
 13. IBM Manual SH20-2095-6: *CADAM APT Interface Installation Guide*, 7th ed. Rye Brook, N.Y.: IBM Corp., 1985.
 14. IBM Manual SH20-3026-3: *CADAM Numerical Control (User Reference Manual)*. Rye Brook, N.Y.: IBM Corp. Dec. 1985.
 15. IBM Manual SH20-2036-1: *Computer-Graphics Augmented Design and Manufacturing system: NC Supplement to User Training Manual*. Rye Brook, N.Y.: IBM Corp. 1982.
 16. IBM Manual SH20-6509-0: *CADAM Interactive User Reference Manual*, vols. 1 and 2. Rye Brook, N.Y.: IBM Corp., 1985.
 17. IBM Manual SC19-6210-2: *CMS User's Guide, Release 3*, 3rd ed. Rye Brook, N.Y.: IBM Corp., 1983.
 18. EIA Standard RS-232-C: *Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*. Washington, D.C.: Electronic Industries Association, Aug. 1969 (reaffirmed 1981).
 19. IBM Personal Computer Professional Series: *Asynchronous Communication Support*, Version 2.0, 2nd ed. Rye Brook, N.Y.; IBM Corp., 1982.
 20. GN6MB-2 *Maintenance Manual*. Elk Grove Village, Ill.: General Numeric Corp.

Kısım IV

IBM APT-AC Nümerik Kontrol İşlemcisine Göre Son İşlemcilerin Tasarımı ve Uyarlanması

Bölüm 11

Son İşlemciler ve CLDATA'ya Giriş

11.1 SON İŞLEMCİNİN GEREKLİLİĞİ

APT-AC NC işlemcisine ek olarak, bir son işlemciye duyulan ihtiyaç, APT programlarının işlenmesinde birçok firmanın ürettiği NC denetleyicilerinin dönüşümü universal NC kodlarına göre yapmamasından kaynaklanır. NC blok formatı tanımlaması ve adres kelimelerinin ifadesi uluslararası standart (ISO 6983/1)¹ ve Amerikan Standartı (EIA RS-274-0)² tarafından tavsiye edilmesine karşın, değişik firmalar bir fonksiyonun tanımlanmasında farklı kodları kullanmışlardır. Sonuç olarak, aynı parçanın NC tezgahında işlenmesi için farklı format ve kodlardaki NC programlarının benzer özellikleri taşınmasına ihtiyaç duyulur. Bundan başka, NC'li tezgahının farklı modellerindeki aynı fonksiyonu tanımlamak için farklı kodları kullanabilir. G50 ve G92 kodlarının FANUC 6T (torna tezgahı) ve FANUC 6MB (freze tezgahı) kontrol sistemlerinde tezgah koordinat sisteminin tanımlanmasında kullanılması tipik bir örnektir. Ayrıca, aynı firmanın aynı modellerindeki denetleyiciler alıcıların istekleri doğrultusunda farklı seçeneklerle donatılır. Sonuç olarak tezgah denetim sistemleri, aynı tip denetleyici olmasına karşın farklı son işlemcilere ihtiyaç gösterir. Bundan dolayı, bir APT programı NC'li tezgahın tanıyabileceği sonuçları verecek NC'li tezgah denetim sisteminin istediği biçimde işlenmelidir.

Bu problemi çözmek için iki yol vardır. Bunlardan birincisi sadece belirli bir tezgah için APT programları oluşturmak işlemcinin tasarlanmasıdır. APT programı oluşturulduktan sonra, arzu edilen NC tezgahının formatına doğrudan dönüştürülebilir. Böyle bir yaklaşım güvenli değildir. Çünkü işlemci içinde yapılan dönüştürme ve hesaplama işlemleri oldukça karmaşıktır. Ayrıca bir NC tezgahı için özel

NC işlemcisi oluşturmak hem güçtür hem de ekonomik değildir. Diğer bir yaklaşımda, kesici konumları ve tezgah işlemleri bilgisini içeren standart bir çıktının oluşturulması amacıyla, hesaplama ve dönüştürme işlemlerini yapan NC işlemcisinin tasarlanmasıdır. Kesici yerleşim bilgisi veya kesici eksen çizgisi bilgisi (kısaca CLDATA *) diye bilinen bu bilgi işlemi ve bilgisayar programı yardımıyla NC işlemcisi bu bilgiyi ilgili formata dönüştürür. SON İŞLEMCI diye bilinen bu kısım kullanıcı tarafından verilecek tezgaha göre APT programını oluşturur. Son işlemci, NC işlemcisi ile karşılaştırıldığında daha küçük bir program, aynı zamanda hem tasarımı hem kontrolü daha kolaydır. Bu yaklaşımda, işlemin ilk safhasında harekete geçirilen NC işlemcisi, birçok çeşit NC tezgahında kullanılacak APT programlarını işlemek için kullanılır. Bu işlem, aynı zamanda kullanıcının ihtiyaç duyacağı tezgah işlemleri haricindeki herhangi bir tezgah için bile son işlemci içermelidir. Bu ikinci yaklaşım, sadece APT için değil diğer yüksek düzey dillerde de NC'li tezgah programlarının hazırlanmasında yaygın olacak kullanılabilir.

Kullanıcı ve tezgah üreticilerinin CLDATA ile kolayca bağlantı kurabilmeleri için, Uluslararası Standart Organizasyonu (ISO) NC işlemci çıktısı için ISO 3592-1978³ ve ISO 4343-1978⁴ isimli iki standart geliştirdi. Bu standartlar herhangi bir dilde yazılmış programların işlemlerini yapan genel amaçlı NC işlemci çıktılarına uygulanmaktadır.

11.2 CLFILE (Center Line File * Kesici Eksen Çizgisi Dosyası *) DOSYASI

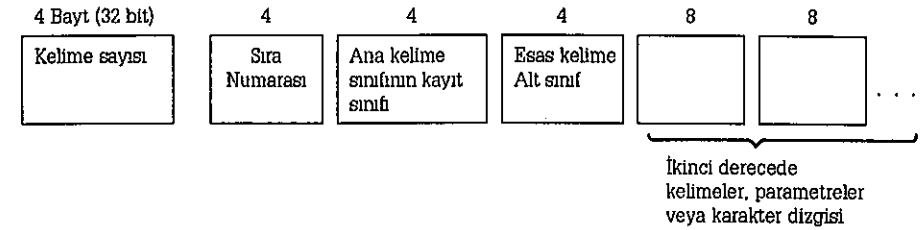
ISO 3592-1978 standardına göre, CLDATA NC işlemcisi tarafından hesaplanan kesici konumunu veya herbirinin bir son işlemci komudunu tanımlayan mantıksal kayıt sıralamasından ibarettir. Mantıksal bir kayıt, her biri bir tamsayı, gerçel sayı veya altı karakterden oluşmuş mantıksal sıralamanın bilemişidir. Kaydın ilk üç mantıksal harfi aynı fiziksel boyuta sahip olup tamsayıdır. Kelimenin diğer kısmı ilk üçle aynı olmamakla beraber, aynı fiziksel boyuta sahiptir. İlk kelime mantıksal kaydın sıra numarasını, 6ncı ise kayıt tipini ifade eder. Kayıttaki diğer kelimeler kayıt tipine bağlı ve kayıt içeriği ile ilgilidir.

ISO standardına göre, 18 kayıt tipi vardır (Tablo 11-1'de verilmiştir). Bu kayıtlar hesaplama aşaması ile ihtiyaç duyulan tüm kayıtları içerir ve NC işlemcisinin dönüştürücü kısmındaki PROFIL isimli çıktı dosyasındadır. Hesaplama bölümünün işleminden sonra, 4000, 7000, 8000 ve 12000 tipi gibi bazı kayıtlar, tezgah işlemi ve kesici hareketinin tanımlanması amacıyla kullanıma ihtiyaç duyulmazlar. Bundan dolayı CLFILE çıktı dosyasında yer almazlar.

ISO standardı mantıksal bir kaydın fiziksel anlamını tanımlamaz. Bunun için farklı kişiler tarafından tasarlanan NC işlemcileri, çıktı dosyasında (CLFILE)⁵ farklı kayıt formatlarına sahip olabilirler.

* IBM-APT AC sayısal Denetim işlemcisinin çıktısı son işlemcinin isteyeceği veriyi ihtiva eden CLFILE'dir.

IBM APT-AC Nümerik Kontrol işlemcisinin işlem sonucu CLFILE'dadır. CLDATA, NC işlemcisinden alınan yazılı form çıktısının genel olarak kullanılan adıdır. IBM-APT-AC NC işlemcisinden alınan CLFILE'in fiziki tanımlaması aşağıdaki gibidir. Kayıttaki ilk üç kelime, sırasıyla kayıt sıra numarasını, kayıt tipi veya ana (esas) kelime sınıfı ve ana kelime alt sınıfını ifade eden 4-bitlik kelimelerdir. Geri kalan ise ikincil kelimeler, parametreler veya deyimdeki karakter işaretlerini ifade eden 8-bitlik kelimelerdir. Her bir mantıksal kayıt son işlemci için ele alınır ve son işlemci tarafından işlenir. IBM APT-AC işlemcisi için ISO standardında (Tablo 11-1) tanımlanan sadece 8 ve 18 kayıt tipleri CLFILE içinde kullanılır. Bunlar (IBM kullanım kılavuzu içinde sınıflanmış) 1000, 2000, 3000, 5000, 6000, 9000, 14000 ve 20000'dir. Bunun yanında, hata teşhis mesajını tanımlayan 13000 kayıt sınıfını da CLFILE içinde yer alır. Mantıksal bir kaydın içeriği, farklı kayıt sınıfları için farklılık gösterir. Bunlar Tablo 11-2 ile 11-9 arasında listelenmiştir. Buna göre, mantıksal bir kaydın kelime sayısı, farklı kayıt sınıflarına göre farklılık gösterir. IBM APT-AC işlemcisinden alınan CLFILE kaydı aşağıdaki formata sahiptir⁵:

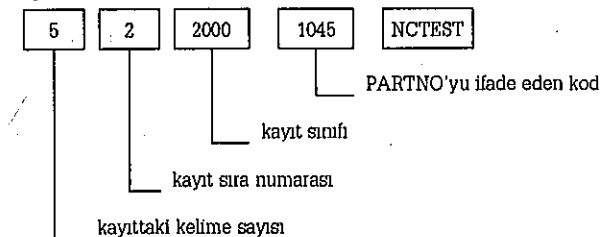


Görüldüğü gibi, o anki kayıttaki kelime sayılarını ifade eden 4 bitlik bir kelime de, herbir kaydın başına eklenebilir. APT esas ve ikinci derecede son işlemci kelimeleri sayılar ile kodlanır (IBM kullanım kılavuzuna bakınız). Örneğin, ARCSLP, MACHIN ve PARTNO son işlemci kelimeleri sırasıyla 1029, 1015 ve 1045 sayılarıyla tanımlanır. İkinci dereceden kelime ON, sınıfı ve alt sınıfını gösteren 58 ve 71 sayıları ile tanımlanır. Böylece, APT yazılımı

PARTNO NCTEST

A0001

şeklinde dir. 2000 kayıt sınıfını ifade eden bu yazılım, aşağıdaki formda CLFILE içinde yer alır:



TABLO 11-1 BİR NC İŞLEMCİSİNİN ÇIKTISINDA BULUNAN KAYIT TIPLERİ

Kayıt Tipi	Kayıt İsmi	İçeriği
1000	Giriş sırası	Orijinal parça programı yazılımının tanımlanması ve sırası
2000	Son işlemci komutları	Son işlemci tarafından işlenen ve sunulan talimatlar
3000	Yüzey bilgisi	Girdi geometrisinin kanonikal biçimi
4000	Bağlı kesici konumu	Sürücü ve parça yüzeylerine bağlı bağlikesici konumu
5000	Kesici konumu	Kesici konumu ve kesici hareket vektörüne bağlı bili
6000	Tolerans ve kesici bilgisi	SD işlemcisini hesaplama kısmı için ihtiyaç duyulan CUT-ONTCUT, kesici tolerans vs. ait bilgiler
7000	Başlangıç hareketi komutu	Başlangıç hareketinde tanımlanmış bir yüzeye bağlı kesici konumu
8000	Hareket bilgisi	Kesicinin son hareketine bağlı kesici yönünü belirtir bilgiler.
9000	Eksen modu; birimler	Eksen birimi veya modlarına bağlı bilgiler (çok eksenli veya üç eksenli)
12000	Özel programlama parametreleri	Özel bir programın çağırılması ile ilgili veri veya bilgiler
14000	FINI kaydı	Para programının durum kaydı
15000	Bölünlenmemiş kesici yolu	Doğrusal olmayan kesici yoluna bağlı bölünlenmemiş bilgi
16000	Kesicinin çevresel tanımı	İş parçası çevresinin tanımlanmasında kullanılan bilgi veya parametreler
17000	Kesici tanımı	Tezgah, malzeme ve kesici tanımları için kullanılan ayrılmış kayıt tipleri
18000	Malzeme tanımı	
19000	Tezgah tanımı	
20000	Son işlemci kayıt tipleri	Son işlemci komutları veya talimatları
28000	Özel kayıtlar	Özel kullanım için ayrılmış kayıt tipleri (standartlaştırılmamış)
32000		

NOT:

- 4000, 7000, 8000 ve 12000 kayıtları son işlemciye girdi olarak kullanılacak CLDATA dosyasının bir parçasını, normal olarak oluşturmaz.
- Bir parça programında kullanılan son işlemci kelimeleri 2000 veya 20000 kayıt tipleri gibi çıktısı iki farklı yolla tanımlanabilir. 2000 kayıt tipindeki bilgiler tam ve gerçek sayıların bileşimi dizgisinden meydana gelir. Yine de 20000 kayıt tipinde karakter dizisi gibi görünür.

* ISO 3592-1978 (3) standardından kısaltılmıştır. Bu tablo Standart Nümerik Kontrollü Tezgahların Mantıksal Yapısı - NC İşlemci Çıktısı - için Uluslararası Standart Organizasyonun izniyle basılmıştır. Amerikan Ulusal Standart Enstitüsü tarafından kullanımı yasaklanmıştır, 1430 Broadway, New York, NY 10018.

1000 kayıt sınıfı (Tablo 11-2) iç deyim numarasını (ISN), APT yazılım etiketini ve girdi kartının 73-80 kolonlarındaki tanımlanan sıralamayı işaret eden her bir deyim için oluşturulur. Böylece, yukarıda verilen durum için iki kayıt; 1000 kayıt sınıfı

6 1 1000 0001 0 A001

ve önceden verilen 2000 kayıt sınıfı oluşturulur.

Bir son işlemci komutu (veya kelime) ve parametreleri 2000 kayıt sınıfı ile ifade edilebilir (Tablo 11-3). Kaydın uzunluğu APT durumu ile değişkenlik gösterir. 2000 kayıt sınıfında 252 parametreye kadar bilgi bulunabilir.

TABLO 11-2 CLFILE içindeki 1000 kayıt sınıfının içeriği

Kelime	İçerik	Kelime Uzunluğu (Bayt)
1nci	Kayıt sıra numarası (tamsayı)	4
2nci	Kayıt sınıf kodu (örnek 1000)	4
3ncü	İç durum numarası (ISN)	4
4ncü	Deyim etkisi (belirtilmemişse 0'dır)	8
5nci	Alfasayısal biçimdeki girdi kartının 73-80 kolonlarındaki tanımlanmış sıralama	8

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Nümerik Kontrol İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtery of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2):

TABLO 11-3 CLFILE içindeki 2000 kayıt sınıfının içeriği

Kelime	İçerik	Kelime Uzunluğu (Bayt)
1nci	Kayıt sıra numarası (tamsayı)	4
2nci	Kayıt sınıf kodu (örnek 2000)	4
3ncü	Son işlemci ana komut (kelime) kodu	4
4ncü	APT bloğunda takimin (1) sağdaki ilk parametre	4
5nci	APT bloğunda taksimin (eğer varsa) ikinci parametre	8
6nci	APT bloğunda taksimin (eğer varsa) ek parametre	8

8 bayt uzunluğundaki her bir komut; en fazla 252 parametre; bu kayıta oluşabilir.

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Nümerik Kontrol İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtery of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2):

3000 kayıt sınıfı (Tablo 11-4) sürücü yüzey bir daire veya silindir ile oluşturulur. Bu kayıt CLFILE içinde 5000 kayıt sınıfı ile birlikte bulunur. 5000 kayıt sınıfı, CLFILE'daki 2000 kayıt sınıfı gibi görülen ARCSLP/ON APT konum tanımlandığı zaman, dairesel enterpolasyon kodunun, G02 veya G03'ü türetilmesi için, parça profilin bağlı kesici konumlarını gösterir.

5000 kayıt sınıfı (Tablo 11-5) bir APT hareket yazılımında tanımlanan kesici konumuyla ilgilidir. GOTO (nokta) veya GODLTA yazılımı ile tanımlanacak bir hareket için, bu kayıta kesici hareketi için yalnızca bir kayıt vardır. Çevresel hareket yazılımıyla doğrusal olmayan bir hareket veya birden fazla kesici hareket ifade eden noktadan noktaya tipi yazılımla kesici hareketini anlatılabilmek için örneğin GOTO/(ızgara); birçok kesici konumu bu kayıta varolabilir. Örneğin C1 çemberi çevresindeki kesici hareketini tanımlayan çevresel hareket yazılımı

TL....GO.../C1... dir.

Girdi kart sırasını tanımlayan 1000 kayıt, C1 çemberinin kanonikal biçimini gösteren 3000 kayıt sınıfı ve kesici konumlarını ifade eden 5000 kayıt sınıfından meydana gelen CLFILE içindeki kayıt takımını türetir.

6000 kayıt sınıfı (Tablo 11-6) APT programındaki tanımlanan kesici ile ilgilidir. Eksen modu, boşluk ve ölçü birimi ile ilgili olan 9000 kayıt sınıfı (Tablo 11-7) hesaplama bölümü için kullanılır. Teşhis mesajları CLFILE içinde 13000 kayıt sınıfı altında saklanır. 14000 kayıt sınıfı ise (Tablo 11-9) programın son bölümünü, FINI bünyesinde barındırır.

TABLO 11-4 CLFILE İÇİNDEKİ 3000 KAYIT SINIFININ İÇERİĞİ

Kelime	İçerik	Kelime Uzunluğu (Bayt)
1nci	Kayıt sıra numarası (tamsayı)	4
2nci	Kayıt sınıf kodu (örnek 3000)	4
3ncü	Altsınıf kodu (CLFILE içinde, bu kelime her zaman iki değere sahiptir. Bu kayıta tanımlanan yüzey sürücü yüzeyidir.)	4
4ncü	Kesici-yüzey ilişki kodu	8
5nci	İşaretleyici yüzey tipi	8
6nci	Kanonikal biçimdeki parametre sayısı	8
7nci	Yüzeyin alfasayısal ismi	8
8nci	Yüzey sembolünün değeri	8
9ncü	Çember (silindir) merkezinin X koordinatı	8
10ncü	Çember (silindir) merkezinin Y koordinatı	8
11nci	Çember (silindir) merkezinin Z koordinatı	8
12nci	Çember (silindir) ekseninin X bileşeni	8
13ncü	Çember (silindir) ekseninin Y bileşeni	8
14ncü	Çember (silindir) ekseninin Z bileşeni	8
15nci	Çemberin (silindir) yarıçapı	8

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtery of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2):

TABLO 11-5 CLFILE İÇİNDEKİ 5000 KAYIT SINIFININ İÇERİĞİ

Kelime	İçerik	Kelime Uzunluğu (Bayt)
1nci	Kayıt sıra numarası (tamsayı)	4
2nci	Kayıt sınıf kodu (örnek 5000)	4
3ncü	Hareket tipini belirten altsınıf kodu = 3 FROM/ yazılımı için 4 GODLTA/ yazılımı için (eğer bu yazılımda yalnız bir hareket vektörü tanımlanmışsa) 5 GOLDTA/ yazılımı için (Birden fazla hareket vektörü tanımlıyorsa) veya GOTO/GO.../ yazılımı için (bir veya daha fazla sonuç noktası oluşturulduğunda kullanılır). 6 5nci baytta anlatılan yazılımların çok noktalı kayıtları için kullanılır (Bu alt sınıf içinde bir seri nokta* oluşabilir).	4
4ncü	FORM/, GODLTA/, GOTO/, veya GO.../ (hiçbirşey yoksa boşluk) yazılımlarında tanımlanmış geometrik özellik sembolü.	8
5nci	4ncü kelimedeki geometrik özellikteki sembolde kullanılan...	8
6nci - 8nci	4ncü kelimedeki noktanın x, y ve z koordinatları veya hareket vektörünün artışı dx, dy, dz bileşenleri.	8
9ncü - 11nci	Kesici eksen vektörünün x, y ve z bileşenleri (çok eksenli programlama yapılırken; herbir sonuç noktası için kayıta bu üç kelime görülür).	8

* Üç eksenli programlama için bu kayıta izin verilen en fazla nokta sayısı 83, çok eksenli programlamada 41'dir. İşlemden sonra, NC işlemcisi üç eksenli programlama için en fazla 81, çok eksenli için ise 41 nokta çıktısı verir.

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtery of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH20-1414-2):

Ayrıca CLFILE içinde 20000 kayıt sınıfı da vardır. CLFILE formatını tanımlayan özel bir APT yazılımı verildiğinde, bu kayıt 2000 kayıdı yerine kullanılır.

APT programına ait bir örnek ve CLFILE içindeki bu programla ilgili kayıtlar Şekil 11-1'deki gösterilmiştir.

11.3 BİR SON İŞLEMİNİN GENEL YAPISI

CLFILE, son işlemde gerçekleştirilen bir dizi işlem sonucunda NC kodlarına dönüştürülür. İzlenen işlem aşağıda açıklandığı gibidir (Şekil 11-2): Son işlemci işleme konduğunda, CLFILE içindeki kayıt son işlemci tarafından okunur. Kayıt öncelikle işlemcinin genel işleme bölümünde değerlendirilir ve sonuçlar bir bellek alanında saklanır. Elde edilen bilgi tekrar değerlendirilir, kontrol edilir ve NC denetçisinin anlayabileceği bir formatta düzenlenir. Sonra son işlemcinin çıktı bölümü ile sonuçlar alınır. Son işlemci bir kaydın işlenmesini bitirdiğinde, tampon dizileri, sayaçlar ve değişkenler bir sonraki kaydın işlenebilmesinde kullanılmak üzere boşaltılır. Sonra ikinci kayıt işleme sokulur. Son işlemci bu döngüyü, FINI kaydının okunup değerlendirmesine kadar devam ettirir.

TABLO 11-6 CLFILE İÇİNDEKİ 6000 KAYIT SINIFININ İÇERİĞİ

Kelime	İçerik	Kelime Uzunluğu (Bayt)
1nci	Kayıt sıra numarası (tamsayı)	4
2nci	Kayıt-sınıf kodu (örnek 6000)	4
3ncü	Kayıt alt sınıf kodu (CLFILE içinde kesici biçimini, boyutunu ve kesici-kontrol yüzeyi ilişkisini ifade eden 6 kullanılır)	4
Aşağıdaki kelimeler kesici tanımlamasına bağlıdır:		
	CUTTER/A1,A2,A3,A4,A5,A6,A7	CUTTER/OPTION,K $\begin{cases} r,h \\ OFF \end{cases}$
4ncü	A1	K = 1 için 1 K = 2 için 1 OFF için 3
5nci	A2	r
6ncı	A3	h
7nci	A4	
8nci	A5	
9ncü	A6	
10ncü	A7	

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtery of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH20-1414-2):

TABLO 11-7 CLFILE İÇİNDEKİ 9000 KAYIT SINIFININ İÇERİĞİ

Kelime	İçerik	Kelime Uzunluğu (Bayt)
1nci	Kayıt sıra numarası (tamsayı)	4
2nci	Kayıt-sınıf kodu (örnek 9000)	4
3ncü	Eksen modunu, boşluk veya ölçü birimini tanımlayan 2,5 veya 9 değerli kod	4
Aşağıdaki kodlar 3ncü kelimedeki koda dayanır:		
	2	5
	(THICK/A1, A2, A3, A4, A5) UNITS	$\begin{cases} INCHES \\ MM \end{cases}$
4ncü	Çok eksenli programlama için 1 Çok eksenli olmayan programlama için 0	A1 INCH'ler için 1.0 MM için 25.4
5nci		A2
6ncı		A3
7nci		A4
8nci		A5

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtery of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH20-1414-2):

TABLO 11-8 CLFILE İÇİNDEKİ 13000 KAYIT SINIFININ İÇERİĞİ

Kelime	İçerik	Kelime Uzunluğu (Bayt)
1nci	Kayıt sıra numarası (tamsayı)	4
2nci	Kayıt-sınıf kodu (örnek 13000)	4
3ncü	Hesaplama bölümündeki hata mesajlarını gösteren kod (3 ve 7 kodları sırasıyla sayısal ve yazılı hata mesajlarını gösterir)	4
Aşağıdaki kelimeler 3ncü kelimedeki koda bağlıdır:		
	3	7
4ncü	Hata mesajı numarası	Yazılı mesajdaki karakter sayısı
5nci		Yazılı mesajdaki ilk sekiz karakter
6ncı		9ncüden 16ncı karakterlere kadar
...		8
...		8
...		8

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtery of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH20-1414-2):

TABLO 11-9 CLFILE İÇİNDEKİ 14000 KAYIT SINIFININ İÇERİĞİ

Kelime	İçerik	Kelime Uzunluğu (Bayt)
1nci	Kayıt sıra numarası (tamsayı)	4
2nci	Kayıt sınıf kodu (örnek 14000) FINI tanımlamasının yapıldığı işaret eder	4
3ncü	0	4

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtery of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH20-1414-2):

Yukarıda açıklanandan açık olarak görünen şu ki, bir son işlemcinin genel olarak aşağıdaki bölümlerden meydana gelmesi gerekir:

1. Hazırlık bölümü. Farklı değişkenler, sayaçlar ve diziler işlemin başlangıcında hazırlanır ve yeni bir işlem döngüsü başladığında güncelleştirilir.
2. Okuyucu bölümü. Bu bölümde, CLFILE içindeki kayıtlar birer birer okunur.
3. Genel veya tezgah-bağımsız işleme bölümü: İki farklı işleme çeşidi uygulamaya konulur: kesici konumlarına bağlı geometrik bilgi hesaplaması ve tezgah işleme komutlarının uygun sayısal kodlara dönüşümü.
4. Özel veya tezgah bağımlı işlem bölümü: Bilgi, NC denetçisinin istediği

APT Program

```

PARTNO NCTEST
MACHIN/GN5CC,9,OPTION,2,0
.....
FROM/P1
GOTO/P2
.....
END
FINI

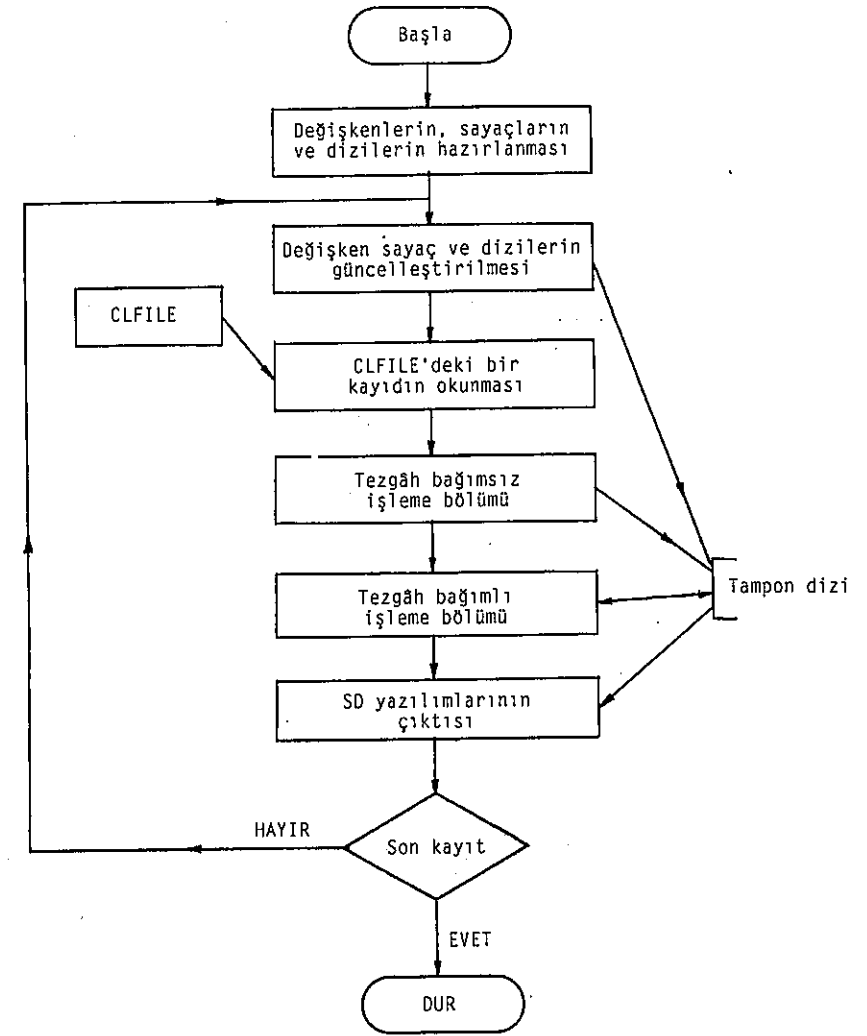
```

CLFILE

6	1	1000	0001	0	0				
5	2	2000	1045						
6	3	1000	0002	0	0				
9	4	2000	1015	GN5CC	9	58 144#	2	0	
.....									
6	i	1000	j	0	0				
9	i+1	5000	3	P1	0	x _{P1} *	y _{P1} *	z _{P1} *	
6	i+2	1000	j+1	0	0				
9	i+3	5000	5	P2	0	x _{P2} *	y _{P2} *	z _{P2} *	
.....									
6	k	1000	m	0	0				
4	k+1	2000	1 ^q						
6	k+2	1000	m+1	0	0				
4	k+3	14000	0						

Şekil 11-1 Bir APT programı ve ilgili CLFILE. Bir kayıt içindeki ilk dört kelime 4 baytlık, geriye kadar 8 byte'lıktır. # - kodu ikincil kelime OPTION * - P1 ve P2 noktaların koordinatları içindir. - kodu sonişlemci kelimesi END içindir.

formata dönüştürülür. Kesici yoluyla ilgili bilginin daha fazla işlenebilmesine bu bölümde ihtiyaç duyulabilir. ayrıca kesici konumlarının, ilerleme miktarlarının ve devir sayılarının kabul edilmiş değerlerin üzerine çıkıp çıkmadığı kontrol edilir.



Şekil 11-2 Son işleminin genel yapısı ve işlem akış şeması.

5. Çıktı bölümü: İşlenen bilginin doğrudan NC tezgahlarında kullanılabilen NC kodları şeklindeki çıktısının (yazılı ve/veya delikli kart/şerit) alınması.

Son işlemcinin tasarımı karmaşık bir işlemdir. Çünkü çok sayıda kelime (komut), parametre ve değişkenin gözönünde bulundurulmasından dolayı içindeki iletişim oldukça karmaşıktır. Bununla beraber, yukarıda sayılan beş bölümün analizi sonucunda, 1, 2, 3 ve 5 numaralı bölümler tezgahla ilgili olmayıp hemen hemen her son işlemcide benzerlik gösterir. Sadece bölüm 4, NC denetçisi ve

tezgahının özelliklerine bağlı olarak tasarlanmış değerlendirme işlemlerini icra eder. Bundan dolayı, eğer mevcut bir son işlemci var ise, yeni bir son işlemcinin tasarımı tezgah-bağımlı işlemi oluşturacak kısmın hazırlanması ile sınırlandırılır (örnek Bölüm 4). Bu husus IBM APT-AC NC işlemcisi ile oldukça basitleştirilmiştir. DAPP (Design Aid for Postprocessor - Son İşlemci için Tasarım Desteği) diye isimlendirilen alt program NC işlemcisinde yer almaktadır¹. DAPP alt programında herhangi bir son işlemciye uygun işlemlerin gerçekleştirilmesini sağlayacak dört bölümü için bir çerçeve mevcuttur. Belirli bir NC tezgahına son işlemci tasarlamak için, kullanıcının NC tezgahına bağlı karakteristik parametreleri hazırlaması ve özel işlemleri icraya koyacak alt programları yazması yeterli olur. Böylece son işlemcinin tasarımı bir hayli basitleştirilmiş olur.

DAPP'da dört işleme bölümü mevcuttur. (Yukarıda açıklanan 1, 2, 3 ve 5nci bölümler). Bunlar Hazırlama ve Düzenleme, Okuyucu, Dağıtma * ve Çıktı bölümleridir. Tezgah-bağımlı işlemi oluşturan bölüme ise [Machine Tool Module (MTM)] tezgah modülü ismi verilir.

11.4 SONİŞLEMCİNİN FONKSİYONLARI

DAPP tabanlı bir sonişlemci, CLFILE dosyasını girdi olarak alan işleyen ve sonra tezgahın okuyabileceği kodda parça programı çıktısı veren bir bilgisayar programıdır. Her son işlemcide olan okuma ve çıktı bölümlerinin yanı sıra bir son işlemci aşağıdaki fonksiyonlara sahip olmalıdır⁸:

1. Parça koordinat sisteminde tanımlanmış bir kesici konumunu tezgah koordinat sistemine dönüştürmelidir.
2. Doğrusal, dairesel, parabolik (eğer kontrolcü bu fonksiyona sahipse) enterpolasyon ve döngü hareketini işleme özelliği olmalıdır.
3. Uygun devir sayısı, ilerleme miktarı, kesici ve farklı tezgah işleme kodlarını oluşturmalıdır.
4. NC tezgahı denetim sistemlerinin ihtiyaç duyduğu komutların oluşumu ve düzenlenmesini yapmalıdır.
5. Kesici yolu ve iş hareketinde, devir sayısında, ilerleme miktarında ve kesicilerin yerleştirilmesinde kontrol ve doğrulama özelliğine sahip olmalıdır. Ayrıca hata durumlarında teşhis üretmelidir.
6. NC tezgahında kullanılan formatta çıktıyı oluşturmalıdır. Bu fonksiyonlar aşağıda kısaca açıklanmıştır.

Bir APT programı yazılırken, programcı, kesici yolunu ve parça geometresini

* Bu terim IBM kullanın kılavuzunda yer almıştır⁷. Burada sadece bilgi olunması amacıyla verilmiştir. Sonraki bölümlerde, bunun yerine GENEL İŞLEME BÖLÜMÜ kullanılmıştır.

en uygun şekilde tanımlayabilmek için genellikle bir koordinat sistemi seçer. Programcının seçtiği, NC'li tezgah kullanıcısı ve ustanın seçtiği ile aynı olmayabilir. Bu koordinat sistemlerinin bağlı konumları veya ilişkisi ORIGIN yazılımıyla tanımlanabilir. Ayrıca birçok kesicinin kullanıldığı bir programda kesici konumlarının dönüşümlerinin yapılmasına ihtiyaç duyulur.

Bir kesicinin diğerine göre konumu Bölüm 6.6.'da tanımlanan aşağıdaki yazılımla belirlenir.

LOADTL/....SETOOL....

Bu dönüşümler NC işlemcisi tarafından yapılmaz. Tezgahtan bağımsız olduklarından, son işlemcinin tezgah-bağımsız işleme bölümünde yer alırlar (Şekil 11-2).

Birçok NC'li torna ve freze tezgahı üç temel hareket biçimine sahiptir. Bunlar noktadan noktaya, doğrusal ve dairesel enterpolasyon hareketidir. Özel çevresel işleme hareketine sahip bazı NC'li tezgahlarda yüksek düzey enterpolasyon hareketi var ise de, bazı tezgahlar sadece noktadan noktaya veya doğrusal enterpolasyon hareketine sahiptir. Torna veya freze tezgahı için kullanılan son işlemci üç temel hareketi oluşturacak yeteneğe sahip olmalıdır. Bu hareketin oluşumu iki adımda gerçekleştirilir: Bunlar tezgahla ilgili olmayan genel işleme, tezgahla ilgili olan özel işlemdir.

Tezgah işlemine bağlı bütün son işlemci kelime ve komutları, uygun kodları verecek şekilde işlenmeli ve dönüştürülmelidir. İşlem belirli bir tezgah için yapılacak ise, kodlar tezgah-bağımlı işleme bölümünde (MTM) yer almalıdır.

Hareket yönü değiştirildiğinde sistemin sonucu olarak kesicinin hareketi köşelerde işin hassasiyetini etkiler. Hareket yönünün değişmesinden önce kesici bilgisinin tanımlanması gerekir. Eski NC tezgahlarında, bu işlem hareketin kısımlara bölünmesi ve kesicinin son kısma geldiğinde ilerleme miktarının azaltılması ile belirlenmekteydi. Modern CNS'li denetleyicilerde bu işlem kesici ilerlemesinin hareket sonuna vardığında yavaşlaması, bir sonraki hareket için hızlandırılmasının otomatik olarak yapılması ile sağlanmaktadır. Artış ve azalma miktarları istenen işleme hassasiyetine göre tekrar verilebilir veya ayarlanabilir. Bundan dolayı, ölçü hassasiyeti son işlemcinin yardımı olmaksızın işin köşesinde oluşturulabilir.

Son olarak, son işlemci bilgisayar kodundaki bilgiyi NC'li tezgah kontrolcüsünün anlayabileceği kod formatına (EIA veya ISO) dönüştürmelidir. Ayrıca çıktı yazılımı (blok) NC denetleyicisinin arzu ettiği formatta olmalıdır. Çıktılar delikli şerit, kart ve çıktı listesi şeklinde kullanıcının istediği bir formda olabilir.

Bu fonksiyonların detayı ileriki bölümlerle ele alınacaktır.

PROBLEMLER

- 11.1 Aşağıdaki yazılım için kaç çeşit CLDATA kaydı oluşturabilir?
GODLTA/1,2,3,2.5
- 11.2 Aşağıdaki yazılım için kaç çeşit CLDATA kaydı oluşturulabilir?
GOTO/1,2,3,2.5
- 11.3 C1 ve LI sembollerinin sırasıyla bir daire ve doğruyu tanımladığını varsayalım. Aşağıdaki yazılım için kaç çeşit CLDATA kaydı oluşturulabilir?
TLRGT, GORGT/C1,TANTO,L1
- 11.4 Bir POCKET (cep) yazılımı için kaç çeşit kayıt oluşturulabilir?
- 11.5 DAPP yaklaşımına bağlı olarak Kısım 11.3'de tanımlanan gibi bir son işlemci tasarlandığını ve CLDATA'yu üç farklı NC'li tezgah için işlediğini düşünelim. Son işlemcinin genel yapısı nedir? Şekil 11-2'de verilen yaklaşımı kullanarak bu son işlemci için bir işlem akış şeması çiziniz.
- 11.6 Eğer sadece bir NC'li tezgah için son işlemci tasarlanırsa DAPP kullanmaksızın işlemcinin genel yapısı nedir?

Bölüm 12

DAPP Tabanlı Son İşlemcideki Temel Değişkenler ve İletişim Kanalları

Bölüm 11.3'de işaret edildiği gibi, DAPP, IBM APT-AC NC işlemcisinde işlemlerin tasarımını kolaylaştırmak için kullanılan bir alt programdır. NC işlemcisi ve DAPP alt programı FORTRAN ve ASSEMBLER dillerinde yazılmıştır⁷. DAPP, tezgah-bağımsız işlemi oluşturacak tam bir işlem akışı hazırlar. Bu işlemde, değişkenlerin, sayaçların ve dizilerin öncelikle hazırlanmasını sağlayan Ön Hazırlık ve Düzenleme Bölümleri, CLFILE'dan kayıtların çıkarımını yapan Okuyucu Bölümü, CLDATA'nın tezgahbağımsız işlemi yapan Genel İşleme Bölümü, işlenen bilgilerin arzu edilen formatta çıktısını almayı temin eden Çıktı Bölümü ve DAPP içinde denetim akışını yöneten Denetim Bölümü mevcuttur. Tezgah-bağımlı işlemi oluşturan Takım Tezgahı Modülü (MTM) kullanıcı tarafından yapılmalı ve temin edilmelidir.

DAPP içinde, bazısı CLFILE'dan alınan bilgiyi ifade eden, bazılarının normal işlem fonksiyonunun algılanmasında DAPP tarafından kullanıldığı birçok değişken vardır. Bu değişkenler FORTRAN COMMON yazılımında tanımlanır ve DAPP içindeki iletişim kanalları gibi kullanılır. Son işlemcinin başlatılmasından önce, hepsinin ilk değerleri tanımlanmalıdır. Bir grup CLFILE kaydının son işleminden sonra, bazı değişkenler tekrar hazırlanmalı veya güncelleştirilmelidir. Değişkenlerin ilk değerleri NC'li tezgah kontrol sisteminin karakteristikleri ve son işlemcinin çıktısına göre kullanıcı tarafından belirlenmelidir.

Bir son işlemci, farklı değişkenlerin ilk değerlerini tanımlayan BLOCK DATA alt programı* ve MTM'nin DAPP ile bağlantısıyla oluşturulabilir.

CLDATA'nın bir setinin okunmasından sonra, bilgi muhtelif tip ve sınıflardaki değişkenler şeklinde DAPP içinde saklanır. Bu değişkenler daha sonra DAPP ile işlendikleri gibi bir bölümden diğerine geçirilirler. Sonuç olarak istenen formatta yazılı veya delikli şerit şeklinde çıktı olarak alınırlar.

Bu bölümde, öncelikle DAPP içindeki temel değişkenler tanıtılacaktır. Sonra CLFILE'ı işlemek için son işlemcinin ihtiyaç duyduğu bilgileri toplamak amacıyla kullanılan metod tanıtılacaktır. Son olarak DAPP'ın değişkenlerini hazırlamak için kullandığı metod tartışılacaktır.

12.1 DAPP'TA KULLANILAN TEMEL DİZİ VE DEĞİŞKENLER

DAPP içinde çok sayıda değişken kullanılmasına karşılık, DAPP tabanlı son işlemciler genellikle beş dizi ve üç değişkene ihtiyaç gösterirler. Bunlar dizi olarak DATBUF, OUTGRD, LOGCAL, LIMITS ve TIME, değişken olarak da OUTCTL, CLKARG ve JLINE'dir. Diğer birkaç tane de sonraki kısımlarda bahsedilecektir. Değişkenler, son işlemcinin doğru işlem yapabilmesine imkan verecek değerleri başlangıçta kazanırlar. Bazılarının bu ilk değerleri, farklı bölümlerdeki CLDATA'nın işleme sonucu olarak değiştirilir. Bundan dolayı, bu değişkenlerin her türlü bilgisi DAPP tabanlı son işlemcide yer alan işlemin anlaşılması için gereklidir.

12.1.1 DATBUF Dizisi

DATBUF dizisi, Genel İşleme Bölümü[†] tarafından işlenen CLDATA bilgisini saklamak için kullanılır. Bu değişkenler daha ileri olarak TTM* tarafından işlenir. Sonuçta DAPP'ın Çıktı Bölümü ile SD kodu şeklinde çıktı alınır. DATBUF dizisi, çift hassasiyet, gerçek-sayı (REAL 8) dizisi şeklinde tanımlanır. Toplam 85 elemanı DATBUF (1)'den DATBUF (85)'e kadar sıralanır. Bu dizinin 1'nciden 71'nci konumuna kadar. Genel İşleme Bölümü MTM veya her ikisi ile yapılan işlemlerin sonucu olarak bir set oluşturulur. Geri kalan DATBUF(72)'den DATBUF(85)'e kadar olanlar MTM'nin kullanımı için ayrılır. DATBUF dizilerindeki değişkenlerin bütün listesi, içerik ve açıklamalarıyla beraber Tablo 12-1'de verilmiştir.

DATBUF(1)'den DATBUF(9)'a kadar olanlar, parmak frezenin ekseninin veya torna kesicilerinin uç yarıçapı merkezinin konumuna bağlı bilginin depolanması için kullanılır. Herbir kesici hareket yazılımı için şunlar söylenebilir: 1, 2 ve 3 konumları, sırasıyla kesicinin X, Y ve Z eksenlerindeki hedef koordinatlarını içerir. 4, 5 ve 6'da başlangıç noktasının X, Y ve Z koordinatları vardır. 7, 8 ve 9 ile

*FORTRAN'daki BLOCK DATA alt programı, işaretlenmiş veya işaretlenmemiş COMMON alanındaki değişkenlerin değerlerini hazırlamak için kullanılır.

[†]Genel işleme rutinleri ve TTM için Bölüm 13 ve 14'e bakınız.

Tablo 12-1

TABLO 12-1 DATBUF DİZİSİNİN İÇERİĞİ

DATBUF(İ)	İçerik	Açıklama
DATBUF(1)	Hedef noktasının x koordinatı	Bu konumdaki koordinat dönüştürüldü ve tezgah koordinat sistemine bağlı olarak verildi.
DATBUF(2)	Hedef noktasının y koordinatı	Yukarıdakinin Aynısı
DATBUF(3)	Hedef noktasının z koordinatı	Yukarıdakinin Aynısı
DATBUF(4)	Başlama noktasının x koordinatı	Yukarıdakinin Aynısı
DATBUF(5)	Başlama noktasının y koordinatı	Yukarıdakinin Aynısı
DATBUF(6)	Başlama noktasının z koordinatı	Yukarıdakinin Aynısı
DATBUF(7)	DATBUF(1) - DATBUF(4)	
DATBUF(8)	DATBUF(2) - DATBUF(5)	
DATBUF(9)	DATBUF(3) - DATBUF(6)	
DATBUF(10)	Programlanmış tezgah ilerleme miktarı, f (birim/dak.) (not 1'e bakınız)	Eğer FEDRAT/IPR, f tanımlanmış ise, ilerleme FEDRAT/ f yazılımdan elde edilir, sonra DATBUF(10) = f/n tanımlanır. n = Devir sayısı M05 sorusuna verilecek cevapla belirlenir. (Tablo 12-6'ya bak)
DATBUF(11)	Hızlı ilerleme miktarı (birim/dak.) (2'nci nota bakınız)	
DATBUF(12)	Hızlı ilerleme anahtarı	DATBUF(12) = 0 Programlanmış ilerleme hızlı değil 1 İlerleme hızlı
DATBUF(13)	İş mili dönme yönü	SPINDLE yazılımdan: DATBUF(13) = 1 Saat ibresi yönünde 2 Saat ibresi tersi 3 Mili durdurma
DATBUF(14)	Devir sayısı, s (dev/dak.)	CLFILE içindeki SPINDL/s, 'den
DATBUF(15)	Kesici kodu, t	CLFILE'deki LOADTL/t'den (3'ncü nota bakınız)
DATBUF(16)	Kesici seçim kodu mf	CLFILE'deki SELECT/t'den (3'ncü nota bakınız)
DATBUF(17)	Sogutma sıvısı kodu	CLFILE'deki COOLNT/... 'den DATBUF(17) = 1 SIVI için 2 PULVARIZE için 3 KILAVUZ için 4 AÇIK 5 KAPALI
DATBUF(18)	Artış miktarında döner tablanın açılma konumu	CLFILE'deki ROTABL 'den (4'ncü nota bakınız)

Tablo 12-1

DATBUF(0)	İçerik	Açıklama
DATBUF(19)	Bekleme zamanı (saniye)	CLFILE'daki CYCLE veya DELAY yazılımından (4'ncü nota bakınız) NC işlemcisi tarafından belirlenir ve CLFILE'da 1000 kayıt sınırı olarak saklanır.
DATBUF(20)	O anki CLFILE'in bilgi setinin iç sıra numarası	
DATBUF(21)	NC blok sıra numarası	
DATBUF(22)	Döner tablının dönüş yönü	SEQNO ile belirlenebilir veya ilk blok için 1'le başlayıp her blok için 1'er artırılabilir. Bu C21 sorusuna 0'dan farklı bir değer verilirse yapılabılır (Tablo 12-7'ye bakınız) ROTABL yazılımı ile ayarlanır: DATBUF(22) = 0 Tanımlanmamış değer. 1 Saat ibresi yönünde (CLW) -1 Saat ibresinin ters yönünde (CCLW) DATBUF(23) = 0 MULTAX/OFF kullanıldığı zaman (kesici eksenini daima Z eksenindedir) 1 MULTAX/ON tanımlandığı zaman (kesici yönü işleme süresince değiştirir) DATBUF(24) = 0 Son işlemci hatası yok (kesici yönü işleme süresince değiştirir) 1 Hata var DATBUF(25) = ANULL Özel işlem yok (5'nci nota bakınız) 9001 END için 9002 OPSTOP için (6'nci nota bakınız) 9003 STOP için (7'nci nota bakınız) 9004 DELAY için 9005 MCHTOL için (4'ncü nota bakınız) 9006 ROTABL için +n AUXFUN/n için (8'nci nota bakınız) -n PREFUN/n için (8'nci nota bakınız) CLFILE'daki MCHTOL/n yazılımından (Tablo 13-5'e bakınız.)
DATBUF(26)	Tezgah-iş kontrol sisteminin dinamik cevaplarının neden olduğu izin verilen işleme hataları.	

Tablo 12-1

DATBUF(0)	İçerik	Açıklama
DATBUF(27)	Kesici ayar kodu	Bu kod CUTCOM/... yazılımı ile belirlenir (4'ncü nota bakınız) DATBUF(27) = 0 CUTCOM/OFF için -1 CUTCOM/LEFT, YZPLAN için -2 CUTCOM/LEFT, ZXPLAN için -3 CUTCOM/LEFT, XYPLAN için -4 CUTCOM/RIGHT, YZPLAN için -5 CUTCOM/RIGHT, ZXPLAN için -6 CUTCOM/RIGHT, XYPLAN için n CUTCOM/n için
DATBUF(28)	İşleme döngüsünü biçimi	CLFILE'daki CYSLE yazılımı ile oluşturulur (4'ncü nota bakınız). İki elemanlı CYCTYP (INTEGER*4) DATBUF(28) 'de kullanıldığı gibi aynı hafıza yerleşiminde tanımlanır. Bu REAL *8 değişkenidir CYCTYP(2) = 82 Delik büyüme (BORE) 153 Delik derinleşme (DEEP) 163 Delme (DRILL) 167 Raybalama (REAM) 168 Kılavuz çekme (TAP) 206 Talaş kırıcı delme (BRKCHP) 255 Havşalama (CSINK)
DATBUF(29)	Döngüdeki en küçük noktanın (z) z koordinatı	Koordinat Genel İşleme Bölümünde paket döngü rutini ile (DLWCYC) hesaplanır (4'ncü nota bakınız).
DATBUF(30)	Döngüde kesicinin geriye çıkma değerinin (z) z koordinatı	Koordinat Genel İşleme Bölümünde paket döngü rutini ile (DLWCYC) hesaplanır (4'ncü nota bakınız).
DATBUF(31)	LOGCAL(4) ile LOGCAL(1)	LOGCAL dizisi (LOGICAL*1) altı elemana sahiptir. LOGCAL dizisi ve DATBUF(31), LOGCAL dizisi anlamında EQUIVALENCE yazılımı ile aynı hafıza yerleşiminde tanımlanır (9'ncü nota ve Bölüm 12.1.3'e bakınız). İlk SD tezgahlarında boşluğun ve delme döngüsünün derinliğinin ayarlanması için ihtiyaç duyulan bilgi tamburların kullanımını sağlar.
DATBUF(32)	Boşluk tambur numarası	İlk SD tezgahlarında boşluğun ve delme döngüsünün derinliğinin ayarlanması için ihtiyaç duyulan bilgi tamburların kullanımını sağlar.
DATBUF(33)	Derinlik tambur numarası	
DATBUF(34)	MACHIN yazılımındaki n3 parametresi (Bölüm 6.12.2 ve 13.4.1.1'e bakınız).	
DATBUF(35)	MACHIN yazılımındaki n4 parametresi (Bölüm 6.12.2 ve 13.4.1.1'e bakınız).	

Tablo 12-1

TABLO 12-1 DATBUF DİZİSİNİN İÇERİĞİ (devam)

DATBUF(i)	İçerik	Açıklama
DATBUF(36)	Dairesel enterpolasyon (DE) düzlemi	Hesaplamaların dairesel enterpolasyon rutiniinde (DLWCIR) yapılmasından sonra Genel İşleme Bölümü tarafından belirlenir (4'ncü nota bakınız). DATBUF(36) = ANULL DE istenmemiş 1 DE X-Y düzleminde 2 DE Y-Z düzleminde 3 DE Z-X düzleminde
DATBUF(37)	Dairesel hareketin yönü	Hesaplamaların dairesel enterpolasyon rutiniinde (DLWCIR) yapılmasından sonra Genel İşleme Bölümü tarafından belirlenir. DATBUF(37) = ANULL ARCSLP/OFF için hareket için 1 ARCSLP/ON ve CLW yönündeki hareket için 2 ARCSLP/ON ve CCLW yönündeki hareket için
DATBUF(38)	Daire parçasının merkezinin x koordinatı	3000 kayıt sınıfından sağlanır ve tezgah koordinat sistemine dönüştürülür.
DATBUF(39)	Daire parçasının merkezinin y koordinatı	3000 kayıt sınıfından sağlanır ve tezgah koordinat sistemine dönüştürülür.
DATBUF(40)	Daire parçasının merkezinin z koordinatı	3000 kayıt sınıfından sağlanır ve tezgah koordinat sistemine dönüştürülür.
DATBUF(41)	Daire parçasının yarıçapı	3000 kayıt sınıfındaki aynıdır.
DATBUF(42)	Dairesel hareketin ilk bölümünde başlangıç noktasından bitiş noktasına kadar dx, dy ve dz artış mesafeleri	Daire parçasının ilk kısmı (1/4 kısım) için bilgi
DATBUF(44)	Daire parçasının başlangıç noktasının yarıçapı	Daire parçasının ilk 1/4 kısmı için bilgi
DATBUF(45)	merkezinine X, Y ve Z düzlemlerinde artış miktarları (I, J ve K)	Daire parçasının ikinci kısmı için bilgi
DATBUF(47)	dx, dy, dz, I, J ve K	
DATBUF(48)		
DATBUF(53)		

Tablo 12-1

DATBUF(54)	'den 'e kadar	Daire parçasının üçüncü kısmı için bilgi
DATBUF(59)		
DATBUF(60)	'den 'e kadar	Daire parçasının dördüncü kısmı için bilgi
DATBUF(65)		
DATBUF(66)	'den 'e kadar	Daire parçasının beşinci kısmı için bilgi
DATBUF(71)		
DATBUF(72)	Ayrılmış	TTM tarafından işleme süresince bilgi saklamak üzere ayrılmış
DATBUF(85)		

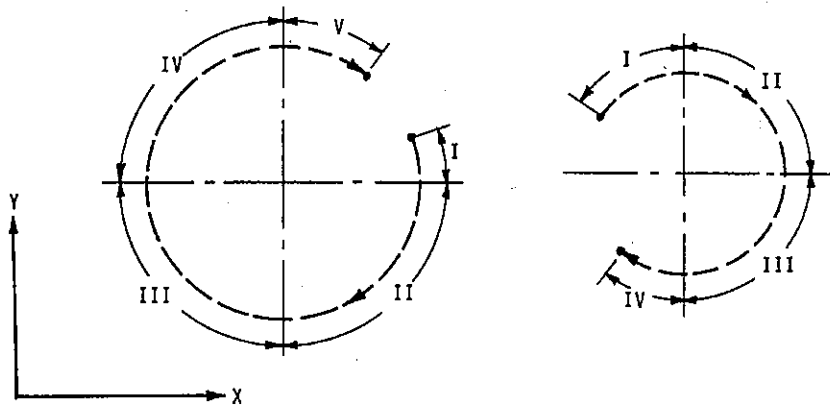
NOTLAR:

- Gerçek birim son işlemcinin tasarımcısı tarafından tanımlanır. Bölüm 12.2.1'deki M30 tezgah tanımlama sorusuna bakınız.
- Modern CNC kontrolcülerinde, hızlı ilerleme miktarı kontrolünün bazı parametreleri ayarlaması ile belirlenir. Bunun yanında, bir NC programı içinde hızlı hareket için ilerleme miktarını tanımlamaya gerek yoktur.
- LOADTL otomatik kesici değişimini hareketi geçiren bir yazıdır. Yine de kesicinin magazindeki yerini ve otomatik kesici değişim komutunun seçimi için SELCTL kullanılır.
- Bölüm 13'e bakınız.
- DAPP'ta, ANULL 9999999'a eşittir ve bir girdinin olmadığı durumlarda özel sembol olarak kullanılır.
- Bir SD programında (yedek durdurma) M01 koduna bağlı APT yazılımı (Bölüm 13'e bakınız).
- Bir SD programında (yedek durdurma) M00 koduna bağlı APT yazılımı (Bölüm 13'e bakınız).
- AUXFUN/n ve PREFUN/n yazılımların sırasıyla M kodlarını ve G kodlarını oluşturmak için kullanılır.
- Şekil 12.4'de verilen BLOCK DATA programındaki EQUIVALENCE yazılımına bakınız.

Otomatik Programlanma Takımı - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3üncü Basım. R'ye Brook, N.Y.: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısıtlanmıştır (SH201414-2):

başlangıç noktasından son noktaya olan X, Y ve Z koordinatlarındaki artış ifade edilir. Bu konumları belirleyen saklanmış koordinatlar tezgah koordinat sistemine dönüştürülür. Bununla NC'li tezgahta kullanılacak X, Y ve Z ile ilgili kodların çıktısı alınabilir. DATBUF(1)'den DATBUF(9)'a kadar olanlar, doğrusal enterpolasyon ve noktadan noktaya hareketin tanımlanmasında gerek duyulan bilginin bir parçasını oluşturur. Bunun yanında, DATBUF(36) ve DATBUF(71) arasında bulunanlar dairesel enterpolasyon hareketi için kullanılır. DATBUF(71) elemanı dairesel enterpolasyonun istenip (ARCSLP/ON) istenmediğini (ARCSLP/OFF) belirtir. Dairesel enterpolasyon (DE*) düzlemi DATBUF(36) ile tanımlanır. DATBUF(38)'den DATBUF(41)'e kadar olan konumlar daire-merkez koordinatları ve dairesel enterpolasyon hareketinin yarıçapına bağlı bilgiyi içerir. APT-AC NC işlemcisi, dairesel kesici yolunu çeyreklere böler. Bölme işlemi, dairenin merkezinden geçen ve koordinat eksenlerine paralel olan doğrularla yapılır. Genel olarak, Şekil 12-1'de gösterilen daire parçasının beş bölümüne kadar oluşum sağlanabilir. Dairesel hareketin herbir bölümü için, başlangıç noktasından bitiş noktasına kadar olan x, y ve z koordinat artışları ve başlangıç noktasından daire merkezine olan x, y ve z değerleri, hareketin ilk bölümü için DATBUF(42)'den DATBUF(47)'ye kadarki konumlarda saklanır. İkinci kısım için DATBUF(48)'den DATBUF(53) arasında yukarıda açıklanan bilgi kayıtları vardır.

DATBUF(28)-DATBUF(33) arasındaki konumlar döngü hareketine bağlı bilgilerin saklanması amacıyla kullanılır. Döngü hakkında bilgiler Bölüm 13'de verilecektir. DATBUF(22) ve DATBUF(18) döner tablanın konum ve dönüş



Şekil 12-1 Dairesel hareketin, merkezinden koordinat eksenlerine paralel olarak geçirilen doğrularla kısımlara ayrılması. Genellikle, beş parçaya ayrılır.

*DE: Dairesel Enterpolasyon (Döngüsünün kısaltması)

yönünü içerir. İşleme toleransına, soğutma sıvısına, kesiciye, devir sayısına ve ilerleme miktarına bağlı işleme parametreleri DATBUF(10), DATBUF(19), DATBUF(26) ve DATBUF(27) içinde saklanır. İşleme modu MULTAX/ON ve MULTAX/OFF, DATBUF(23)'e kayıt edilir. 34 ve 35 DATBUF dizi elemanları, bir APT programındaki MACHIN yazılımında tanımlanmış parametrelerin kabulü için kullanılır. Yine de NC çıktısı ve CLDATA'daki kayıt sıra numarasını 20 ve 21'nci elemanlarda bulunur. DATBUF(24) değişkeni, son işleme süresince meydana gelecek hataları tespit eder.

DATBUF(1)'den DATBUF(71)'e kadarki elemanların içeriği, Genel İşleme Bölümünde hazırlanır ve işlenir, daha sonra MTM tarafından tekrar yerleştirilir. Kaynak bilgisinin saklanması için DATBUF dizisinin yukarıda açıklanan bilgileri kullanılmadığı NC çıktı kodları 72'den 85'nci elemanların kullanımı ile sağlanır.

DATBUF dizisi, eski NC ve yeni BSK denetleyicilerini içeren değişik tasarım denetleyicilerinde son işlemcilerin kullanımı için tasarlanır. Elemanlardan bazıları modern CNC'li sistemler için son işlemci tasarımında kullanılmayabilir. Örneğin, FANUC 6MB denetleyicisi için kesicinin hızlı hareketini tanımlayan yazılımda hareket hızının belirtilmesine gerek yoktur. Böylece DATBUF(11) kullanılmaz. Ayrıca modern CNC'li tezgahlar, işlemede kesici konumlarını kontrol amacıyla tambur kullanmadıklarından, DATBUF(32) ve DATBUF(33)'de kullanılmayan elemanlardır.

DATBUF dizisindeki bazı elemanlar, CLDATA kayıt bilgi setlerinin işleminin tanımlanmasından sonra, yeni bir set işleminin başlamasından önce, DAPP'ın Düzenleme Bölümü ile hazırlanır.

12.1.2 OUTGRD Dizisi

OUTGRD, toplam 200 elemana sahip 20 x 10 iki boyutlu bir dizidir. Elemanlar 4-bayt tamsayılar (INTEGER*4) şeklinde tanımlanır, ve yirmi gruba bölünebilir.

OUTGRD(i,1), OUTGRD(i,2),..., OUTGRD(i,10) (i= 1, 2, ..., 20)

Herbir grubun 10 elemanı vardır. Gruptaki 1'den 9'a kadar olanlar, OUTGRD(i,1), ..., OUTGRD(i,9), NC kelimesinin çıktı formatını tanımlamak için kullanılır. Onuncu eleman, OUTGRD(i,10), bilginin tanımlanan formattaki NC kodu şeklindeki çıktısının DATBUF dizisindeki konumunu belirtir. Böylece, eğer OUTGRD(3,10) 1 gibi hazırlanmışsa, DATBUF(1)'deki saklanan bilgi, OUTGRD(3)'den OUTGRD(3,9) arasında tanımlanan formatta çıktı olur. Bu gruptaki değişik konumların kullanımı ve içeriği Tablo 12-2'de listelenmiştir. 20 SD kelimesi ve formatları DAPP içinde tanımlanabilir.

Tablo 12-2

TABLO 12-2 OUTGRD ELEMANLARININ HER BİRİNİN (i) İÇERİĞİ

Eleman	İçerik	Delikli şerit (DŞ) veya yazılı çıktı (YÇ) kullanımı	Cevaplara bağlı sorular*
OUTGRD(i,1)	Kelime adres formatı için harf (N,G)	DŞ ve YÇ	C09
OUTGRD(i,2)	= 0 Herbir NC kodunu oluşturan tablo kodu yok = n Herbir kelimeyi oluşturan tablo kod numarası NC kelimesindeki ondalık dijiti sayısı = n NC kelimesindeki toplam dijiti sayısı = Negatif değer Alfa Sayısal anlamlı olan çıktı = 0 veya 1 yazılı çıktıda gözardı edilecek ondalık nokta	DŞ ve YÇ DŞ ve YÇ DŞ ve YÇ DŞ ve YÇ	C10 C12 C11
OUTGRD(i,3)	2 Ondalık kısım yazılacak	YÇ	C13 ve C15
OUTGRD(i,4)	-1 Ondalık kısım gözardı edilecek ve pozitif değer için + işareti yazılacak		
OUTGRD(i,5)	-2 Hem ondalık kısım hemde + işareti yazılacak		
OUTGRD(i,6)	Yukarıdan aynı (Bununla beraber bu konum delikli şerit çıktının formatını tanımlamak için kullanılır)	DŞ	C14 ve C16
OUTGRD(i,7)	= 0 Çıktıda yok edilecek sıfırlar	DŞ ve YÇ	C17 ve C18
OUTGRD(i,8)	1 2 3 Kolon numarası, çıktıda kelimenin nereden başladığına işaret eder.	YÇ	C19
OUTGRD(i,9)	= 0 Sadece bu NC kelimesi yazılacak	DŞ ve YÇ	C20
OUTGRD(i,10)	= n DATBUF dizisindeki konum numarası. DATBUF(n), SD kodunda çıktı alınacak bilgiyi içerir.	DŞ ve YÇ	MTM tarafından ayarlanır.

ÖRNEK: FANUC 6MB kontrolünde, G kodları için OUTGRD(i,j) (j=1,...,9) elemanlarını aşağıdaki gibi tanımlayabiliriz:

OUTGRD(2,1) = G; OUTGRD(2,2) = 0; OUTGRD(2,3) = 0; OUTGRD(2,4) = 2; OUTGRD(2,5) = 1;
OUTGRD(2,6) = 1; OUTGRD(2,7) = 0; OUTGRD(2,8) = 6; OUTGRD(2,9) = 1

G kodları gibi sadece 2 basamak içeren tanımlanmış çıktı formatı kesinti dijitsiz ve tablolaştırılmış kodsuzdur. Ondalık noktaya da ihtiyaç duyulmaz. (+) işareti ne delinir ne de yazılır. G koddaki sıfırlar atılmaz. Bu kelimenin yazımı 6'ncı kolonda başlar. Kod hem delinip hem de yazılacaktır. Böylece eğer OUTGRD(2,10), 73 gibi belirlenir ve DATBUF(73) = 1 olursa, çıktı G kodu G01 olur.

*Kısım 12.2.2 ve Tablo 12-7'ye bakınız. ¹ Çevirenin kısıtlaması

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım, R'ye Brook, N.Y., IBM, 1985

(Courtesy of International Business Machines Corporation) IBM ile yazılım hakkındaki kısıtlamadır (SEP2014142)

Grup içindeki 1'den 9'a kadarki elemanların içerdiği değer ve kelimeler, DAPP Soru Bankası diye isimlendirilen sorulara verilen cevaplar ile son işlemci tasarımcısı tarafından hazırlanır. Sorulara verilen cevap değerleri, son işlem sürecince değişmeden tutulur. 10'uncu elemanın değeri ise değiştirilip, MTM ile ayarlanabilir. Böylece, DATBUF dizisinin belirli konumundaki saklanmış bilgi seçilip istenen formatta çıktısı alınabilir.

DATBUF dizisi içinde saklanan bilgi sayısal veya alfa sayısal olabilir. Sayısal bir değerın çıktısı alınırken, Çıktı Bölümünün, çıktı formatını belirleyen dokuz elemanın, OUTGRD(i,1)'den OUTGRD(i,9)'a kadar, içerdiği bilgiye ihtiyaç duyulur. Çıktı alfa sayısal ise, sistem sadece OUTGRD(i,4) ve OUTGRD(i,8)'deki bilgiye ihtiyaç gösterir.

12.1.3 LOGCAL Dizisi

LOGCAL, altı elemanlı LOGICAL*1 dizili bir dizidir. 1'den 4'e kadar olan elemanlar tezgah işlem döngülerinin işlenmesi için kullanılır ve içeriği aşağıdaki gibidir:

LOGCAL(1) = .TRUE.	Z doğrultusunda kesici konumu değişimi döngüde meydana gelir.
.FALSE.	Z doğrultusundaki kesici konum değişim döngüde meydana gelmez.
LOGCAL(2) = .TRUE.	Kesici değişimi derhal takip eder.
.FALSE.	Kesici değişimi derhal takip etmez.
LOGCAL(3) = .TRUE.	Derin delik delme döngüsü etkili (BRKCHP veya DEEP)
.FALSE.	BRKCHP veya DEEP döngüsü etkili değil.
LOGCAL(4) = .TRUE.	BRKCHP veya DEEP döngüsünde, son z konumuna ulaşılr.
.FALSE.	BRKCHP veya DEEP döngüsünde, son z konumuna ulaşılmaz.

Eğer bir CYCLE yazılımı* tanımlanursa, LOGCAL(1), .TRUE.'ya ayarlanır. Eğer CYCLE/OFF yazılımı seçilirse, LOGCAL(1), .FALSE.'a dönüşür. Eğer ikinci yazılım LOADTL ise, döngüdeki LOGCAL(2), .TRUE.'dur. Bu değişken, kesicinin Z konumunun ayarlanması amacıyla Genel İşleme Bölümü tarafından kullanılır. LOGCAL(3) ve LOGCAL(4), derin deliklerin gagalama delme döngüsünde kullanılır. Bu dört konumun bilgisi, Genel İşleme Bölümündeki paket döngü işlemi, DLWCYC (Bölüm 13'e bakınız) ve DAPP ana işlemi ile ayarlanır. Diğer iki konum, MTM tarafından kullanılması için ayrılr.

* Bölüm 13'e bakınız.

12.1.4 LIMITS Dizisi

LIMITS, 11 elemanlı bir real*8 dizisidir (Tablo 12-3). Bunlardan 8 tanesi, kesici hareketinin X, Y ve Z eksenlerindeki sınırlarını, en büyük ilerleme miktarını, ve kontrol çözünürlüğünü (veya herbir koordinat ekseninde en az artışı) saklamak için kullanılır. Bu sınırlar DAPP Soru Bankasındaki soruların cevaplanması ile son

TABLO 12-3 ONBİR ELEMANLI LIMIT DİZİSİ (REAL*8)

Sınırlar(i)	İçerik	Cevaplara Bağlı Soru* Seti
LIMITS(1)	En büyük x koordinatı	M01
LIMITS(2)	En büyük y koordinatı	M02
LIMITS(3)	En büyük z koordinatı	M03
LIMITS(4)	Ayrılmış	
LIMITS(5)	Ayrılmış	
LIMITS(6)	Kontrolcü çözünürlüğü veya en küçük artış koordinatı	C04
LIMITS(7)	Ayrılmış	
LIMITS(8)	En büyük ilerleme miktarı	M06
LIMITS(9)	En küçük x koordinatı	M01
LIMITS(10)	En küçük y koordinatı	M02
LIMITS(11)	En küçük z koordinatı	M03

*Kısım 12.2.1 ve 12.2.2 ve Tablo 12-6 ile 12-7'ye bakınız.

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2):

işlemci tasarımcısı tarafından belirlenir. Genel İşleme Bölümü ve MTM tarafından hareketin sınırlarının veya hızın artırılmasının sağlanması amacıyla kullanılır. LIMITS(6) elemanı, hesaplama sonuçlarının yuvarlatılması için Genel İşlem Bölümüne kullanılır.

12.1.5 TIME Dizisi ve CLKARG Değişkeni

TIME dizisi sekiz elemana (REAL*4) sahiptir. TIME(2)-TIME(6) arası, tezgah işleminin ihtiyaç duyduğu zamanla ilgilidir. Bunlar SD'li tezgah kontrol sisteminin karakteristiklerine bağlı son işlemci tasarımcısı tarafından belirlenmelidir. TIME(1) elemanı, o anki işleme durumu ile tanımlanmış kesici yolunun tanımlanması için kullanılır. Ayrıca kesici seçimi veya bağlanması için gerekli bekleme zamanının belirlenmesi amacıyla da kullanılır (eğer SELECTL, LOADTL ve DELAY yazılımı tanımlanmış ise). TIME(1), Genel İşleme Bölümündeki alt yordamlar olan Geometrik ve Yardımcı bölümler ile hesaplanır ve hazırlanır. Dizinin 7'nci ve 8'nci elemanları MTM'nin kullanımı için ayrılır.

Tablo 12-4'de TIME dizisindeki değişik elemanların içeriği verilmiştir.

TABLO 12-4 SEKİZ ELEMANLI TIME DİZİSİ (REAL*4)

Time(i)	İçerik	Açıklama
TIME(1)	O anki (güncel) blok için işleme zamanı	Geometrik ve Yardımcı Bölümler tarafından yerleştirilir. Zaman, doğrusal, dairesel ve döngü işleme hareketi yordamları ile hesaplanır. Sonra, eğer DELAY, SELECTL veya LOADTL yazılımı belirlenirse, bu durumdaki zaman Yardımcı Bölüm ile güncelleştirilir. Detaylı bilgi için Bölüm 13'deki kısımlara bakınız.
TIME(2)	Bir bloğun yazılımı için en az zaman	DAPP Soru listesindeki C05 sorusuna bağlı cevapla yerleştirilir.
TIME(3)	Döner tablanın 360° döndürülmesi için gerekli zaman	M12 sorusuna verilen cevapla yerleştirilir.
TIME(4)	İş milini başlama ve durdurma zamanı	M21 sorusunun cevabı ile belirlenir ve yerleştirilir.
TIME(5)	Kesicinin bağlanma zamanı	
TIME(6)	Kesici seçimi için zaman	M22 sorusunun cevabıyla yerleştirilir.
TIME(7)	TTM'nin kullanımı için ayrılmıştır	M23 sorusunun cevabıyla yerleştirilir.
TIME(8)	TIME(7)'nin aynısıdır	

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2):

CLKARG değişkeni, o an kullanılacak NC bloğunun (yazılım) durumu için gerekli zamanı tanımlayan REAL*1 değişkenidir. Bu TTM tarafından yerleştirilir ve o anki NC bloğunda tanımlanan işlem(ler)in oluşumunda ihtiyaç duyulan ve kesici hareketinin sonu için gerekli olan zaman miktarı değeridir. CLKARG değişkeni, Çıktı Bölümünde kullanımından sonra sifıra eşitlenir.

TABLO 12-5 OUTCTL DEĞİŞKENİ VE SONUÇ HAREKETİ İLE KABUL EDİLEBİLEN DEĞERLER

OUTCTL	Sonuç Hareketi
1	Okuyucu bölümünün bir set yeni bilgiyi CLFILE'dan okuması, kontrolün Değerlendirme Bölümüne dönmelerinden sonra yapılır.
4	MTM çalıştırılır.
5	Birleştirilmiş işleme zamanı, iç sıra numarası (ISN), bir NC bloğunun MTM tarafından işlenmiş bilgisinin yazılı çıktısı alınır.
6	İç sıra numarası (ISN) ve bir NC bloğunun MTM tarafından işlenmiş bilgisinin yazılı çıktısı alınır.
7	Bir NC bloğunun MTM tarafından işlenmiş bilgisinin çıktısı alınır.
10	Son işleme yapılır.
11	Delikli şerit çıktısı
12	Sadece "*****" çıktısı alınır.

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2):

12.1.6 OUTCTL Değişkeni

OUTCTL (INTEGER*4) değişkeni, arzu edilen çıktının seçimi ve DAPP içinde kontrol akışını yönetmek için MTM tarafından kullanılır. Değişken ve kabul edilen değerlerinin içeriği Tablo 12-5'de verilmiştir. İşleme süresince farklı değerler MTM tarafından seçilirse, sonuç olarak farklı çıktı işlemi ve kontrol akışı meydana gelir. Değişkenin değeri, MTM'nin belirli bir işleme safhasındaki kontrol istekleri ve çıktısına bağlı olarak MTM'nin süresince seçilmelidir.

12.1.7 JLINE Değişkeni

INTEGER*4 değişkeni, JLINE, o anki çıktı sayfasındaki satır numarasını tanımlar. Değişken bir sayfadaki satır sayısını kontrol için kullanılır. DAPP'ın Çıktı Bölümü ile herbir satır için 1'er olacak şekilde otomatik olarak artırılır. Eğer tasarımcı, DAPP'ın Çıktı Bölümünü kullanmaksızın MTM'de bir mesaj yazdırmak isterse (Örneğin, FORTRAN'ın PRINT komudunu kullanarak) ilgili yazılım:

JLINE = JLINE + N

olur. Bu yazılım, N adet satırın yazımını sağlayacak, N'nin JLINE değerini artırması için MTM'ye eklenir.

12.2 DAPP TABANLI SON İŞLEMCİDEKİ DEĞİŞKENLERE İLK DEĞERLERİN ATANMASI: DAPP SORU LİSTESİ

Her programın işlenmesinde olduğu gibi, işlemin başlamasından önce bütün değişkenlerin bir ilk değer alması sağlanmalıdır. Bir son işlemcinin CLFILE'daki çok farklı bilgiyi işlenmesinde, işleme isteklerine bağlı olarak ilk değerde yüklenmiş değişken sayısı oldukça fazladır. Bu ilk değerler aşağıdakilere bağlı bilgileri içerir.

1. NC'li tezgah (maksimum programlanabilir ilerleme miktarı, döner tabla, zamanın seçimi ve kesicinin bağlanması)
2. Çeşitli hareket tiplerinin kontrolünü sağlayan, herbir ekseninde en az artışa ve girdi program formatına sahip NC kontrolcüsü
3. Delikli şerit formatı ve yazılı çıktı gibi özelliklerin son işlemci çıktısında olması

Bir son işlemci tasarımcısı, ilk değer ataması yapılacak tüm değişkenleri ve bu ilk değerlerin tanımlanmasında ihtiyaç duyulacak bilgiyi toplamak zorundadır. Bu, özellikle NC'li tezgahlar ve son işlemci tasarımı konusunda yeterli bilgi sahibi olmayanlar için oldukça güç bir işlemdir.

Değişkenlerin ilk değerlerinin atanmasında gerekli bilginin toplanması için, DAPP, DAPP Soru Listesi diye isimlendirilen bir soru listesine sahiptir. Bu soruların cevaplandırılması ile, tasarımcı ilk değer atayacağı değişkenleri bulur ve ilk değerlerini belirler. DAPP ayrıca QUEST ismi verilen ve ilk değer atama işlemini yapan FORTRAN BLOCK DATA alt programını oluşturacak bir programa sahiptir.

DAPP soru listesinde sorular dört grupta listelenir. Bunlar tezgah, NC ve CNC kontrolcüsü, delikli şerit ve yazılı çıktı için gerekli şartları içerir.

12.2.1 Tezgah Tanımlama Soruları

Tezgah tanımlama grubundaki 30 soru (Tablo 12-6) NC'li tezgahın karakteristikleri ve parametrik ayarı ile ilgilidir. Bu sorulara verilen cevaplar, CLFILE'in işlenmesi için son işlemci tarafından ihtiyaç duyulan tezgah parametrelerini oluşturur. Tablo 12-6'daki sorular, eski ve yeni tasarımlı NC'li tezgahlar içindir. Bir soru cevaplanır veya cevaplanmaz. Son durumda, verilen cevap DAPP tarafından kullanılacak olandır. Bir cevap bir veya daha fazla sayısal değer veya karakter olabilir. Cevaplarla belirlenecek dizinin elemanları veya değişkenler Tablo 12-6'nın son kolonunda listelenir. Herbir soru cevaplanmadan önce, dikkatlice üzerinde durulmalıdır.

Örneğin, M01, M02 ve M03 sorularına verilen cevaplar, LIMIT dizisinin; LIMIT(1)'den (3)'e ve (9)'dan (11)'e kadar; belirli elemanlarının değerlerinin tesbiti için kullanılır. DAPP'ın Geometri Bölümü, programlanmış koordinatları bu sınırlarla (limits) karşılaştırır. Eğer bu sınırları aşarsa, hata mesajı oluşur. Bununla beraber, birçok NC'li tezgahta, tezgah kontrol sisteminin sıfır noktası çalışma alanının herhangi bir yerinde belirlenebilir. Böylece, gerçek sınır konumunu temsil eden en büyük ve en küçük x, y ve z koordinatları, koordinat sisteminin sıfırına (orijin) bağlıdır. M01, M02 ve M03 sorularına verilen cevaplar genellikle Şekil 12-2'de belirlenmiş sıfır noktasındaki uygulama alanına dayanılarak hazırlanır.

Örneğin, sıfır noktasının A noktasında, tanımlanmış B konumunun koordinatlarının da (x, y) olduğunu varsayalım. Tanımlanmış konum; $x < x_{max}$, ve $y < y_{max}$ olsa bile; izin verilen kesici hareket alanını aşabilir. Bundan dolayı, verilen cevaplar genellikle, bir tezgah işlemi için kesici hareketindeki gerçek sınırlar değildir.

M04 sorusundaki hızlı ilerleme miktarına, değerinin hızlı hareket yazılımında tanımlanması gereken eski NC'li tezgahlarda ihtiyaç duyulur. Kısım 12.1.1'de işaret edildiği gibi, modern CNC kontrolcülerde ihtiyaç duyulmaz.

M09'dan M14'e kadar olan sorulara döner tablayı tanımlamak için ihtiyaç duyulur. Eğer NC 'li tezgahta döner tabla yok ise, bu sorular geçilir.

M15'den M18'e kadarki sorular, NC kontrolcüsünün tasarım karakteristiklerine göre cevaplanır. CNC kontrollü birçok freze tezgahı üç koordinat ekseninde

Tablo 12-6

Soru numarası (1'nci nota bakınız)	Soru	Cevap (2'nci nota bakınız)	Varolan cevap(lar)	Birim	Cevap(lar)la ayarlanan değişken(ler) (3'ncü nota bakınız)
M01	En büyük ve en küçük x koordinatı nedir?	x_{min}^{\dagger} x_{max}^{\dagger}	-99999.99 +99999.99	†	LIMITS(9)
M02	En büyük ve en küçük y koordinatı nedir?	y_{min}^{\dagger} y_{max}^{\dagger}	-99999.99 +99999.99	†	LIMITS(10)
M03	En büyük ve en küçük z koordinatı nedir?	z_{min}^{\dagger} z_{max}^{\dagger}	-99999.99 +99999.99	†	LIMITS(11)
M04	Hızlı ilerleme değeri	f_1^{\dagger}	1.0	in/dak veya mm/dak.†	LIMITS(3) DATBUF(11)
M05	İptal edildiğinde hızlı ilerlemenin etkisi hala sürecektir mi?	YES/NO	NO		IRPMOD (INTEGER*4) = 1 YES = 0 NO
M06	Programlanabilir en büyük ilerleme değeri?	f_1^{\dagger}	99999.99	in/dak veya mm/dak.†	LIMITS(8)
M07	Eğer belirli değilse, kesici ilerleme değeri? (Örnek varolan değer)	f_2^{\dagger}	1.0	in/dak veya mm/dak.†	DATBUF(10)
M08	Varolan devir sayısı değeri?	s^{\dagger}	1.0	dev/dak	DATBUF(14)
M09	Tezgaha döner tabla var mı? Eğer cevap YES ise, tablaya yapılabilecek hareket sayısını, m , giriniz.	YES/NO m^{\dagger}	NO 1	derece	TABLGO (4'ncü nota bakınız) TPLIMIT (REAL*8) = (360/m).10 ⁶
Eğer M09'ncü soruya cevap NO ise M10'dan M14'e kadar olan sorular ihmal edilebilir.					
M10	Döner tablanın dönüş yönü nedir?	CLW/CCLW/HER İKİSİ			TABLGO
M11	Hangisi döner tablanın pozitif dönüş yönüdür?	CLW/CCLW	CLW		TABLGO
M12	Döner tablanın 360 dönüşü için kaç saniye gereklidir?	s^{\dagger}	1.0	saniye	TIME(3)
M13	Tezgah koordinat sistemine göre, tablanın merkezinin koordinatları nedir?	x^{\dagger} y^{\dagger} z^{\dagger}	0 0 0	† † †	CORTRS(1) (REAL*8) CORTRS(2) CORTRS(3)

Tablo 12-6

Soru numarası (1'nci nota bakınız)	Soru	Cevap (2'nci nota bakınız)	Varolan cevap(lar)	Birim	Cevap(lar)la ayarlanan değişken(ler) (3'ncü nota bakınız)
M14	GOHOME komutu ıncı edildiğinde, döner tab sıfır noktasına döner mi?	YES/NO	NO		IGOTAB (INTEGER*4) = 1 YES = 0 NO
M15	Tezgahın X, Y ve Z eksenlerindeki hareketi hızlı mı?	YES/NO	NO		SIMOSW(4) (INTEGER*4) = 1 YES = 0 NO
M16	Tezgahın X, Y eksenlerindeki hareketi hızlı mı?	YES/NO	NO		SIMOSW(1) = 1 YES = 0 NO
M17	Tezgahın X, Z eksenlerindeki hareketi hızlı mı?	YES/NO	NO		SIMOSW(2) = 1 YES = 0 NO
M18	Tezgahın Y, Z eksenlerindeki hareketi hızlı mı?	YES/NO	NO		SIMOSW(3) = 1 YES = 0 NO
M19	Tezgah eksenleri ile bağlı olduğu eksenler arasındaki ilişki APT'de nedir? (5'nci nota bakınız)	+1 veya -1 (X için) +1 veya -1 (Y için) +1 veya -1 (Z için)	+1 +1 +1		CONVEN(1) (REAL*8) CONVEN(2) CONVEN(3)
M20	Tezgah için sıfır noktası gerekli mi? Eğer cevap YES ise sıfır noktasının (z_1, y_1, z_1) koordinatları tanımlanmalıdır? (6'nci nota bakınız)	YES/NO x_1^{\dagger} y_1^{\dagger} z_1^{\dagger}	0 0 0	† † †	IGOHOM (INTEGER*8) = 1 YES = 0 NO
M21	İş milinin başlatma ve durdurma zamanı?	t_1^{\dagger}	0	†	HOME(1) (REAL*8)
M22	Kesiciyi bağlama zamanı?	t_2^{\dagger}	0	†	HOME(2)
M23	Kesiciyi seçme zamanı?	t_3^{\dagger}	0	Saniye	HOME(3)
M24	İş milinin yönü: Yatay (Y) Düşey (D)	t_4^{\dagger} t_5^{\dagger} Y/D	0 0 D	Saniye Saniye Saniye	TIME(4) TIME(5) TIME(6) IZHOR (INTEGER*4) = 1 Y için = 0 D için

Tablo 12-6

M25	Tezgah işleme döngüsü oluşturabilir mi?	YES/NO	NO	DRUMZE (INTEGER*4) = 0 NO için 1 YES için
M26	Tezgah paket döngüler için tambur, kam veya dayama kullanılıyor mu?	YES/NO	NO	ZMOD (INTEGER*4) = 0 NO için 1 YES için
M27	Eğer cevap YES ise, tamburdaki boşluğu veya kam numarasını giriniz?	$n_1^†$	0	R10 (INTEGER*4) Z10 (INTEGER*4)
M28	Tambur veya kam numarasının başlangıç derinliği?	$n_2^†$	0	CLDRMS (INTEGER*4)
M29	Tambur veya kam boşluk numarası?	$n_3^†$	0	DPDRMS (INTEGER*4)
M30	Kam veya tamburların derinlik numarası? Tezgah hangi ölçü birimini mm. veya inç kullanıyor?	$n_4^†$ IN veya MM	0	PPUNIT (REAL*8) ve QNUNITT (REAL*8) = 1 INÇ için 25.4 MM için

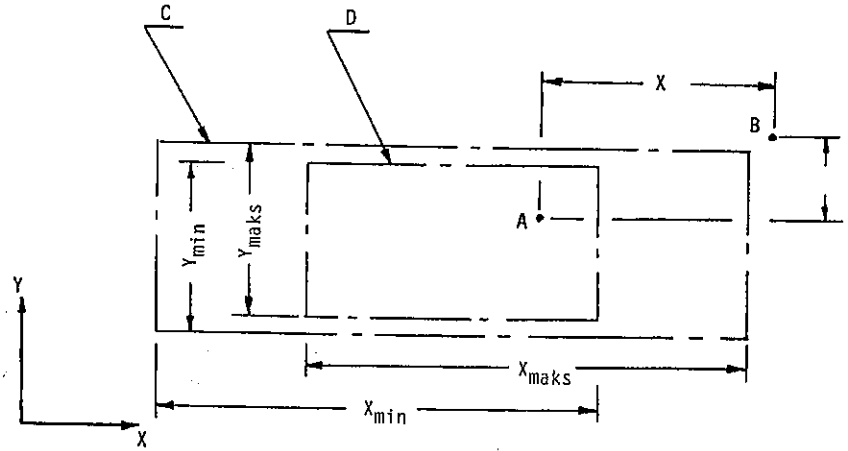
NOTLAR:

- Girdi kartında soru numarası 1'nci ile 3'ncü kolonlar arasında tanımlanmalıdır.
- Girdi kartının 5 ile 72 kolonları arasında herhangi bir yerde tanımlanabilir. Eğer soru birden fazla cevaplı ise, cevaplar aynı satırda, aralarına boşluk bırakılarak verilmelidir.
- Herhangi bir açıklama verilmemiş ise, bu kolonda verilen değişken cevabın değeri olarak kabul edilir.
- TABLGO, INTEGER*4 değişkenidir. Değeri aşağıdaki tanımlandığı gibidir:
TABLGO = 0 Döner tablo yok.
1 Tablo sadece CLW dönüşüne sahiptir
-1 Tablo sadece CCLW dönüşüne sahiptir.
+3 Her iki yönde de dönüşe sahip olup, dönüşün pozitif yönü CLW'dir.
-3 Her iki yönde de dönüşe sahip olup, dönüşün pozitif yönü CCLW'dir.
- Bazı NC'li tezgahlarda, kesici pozitif bir ölçünün işlenmesi için negatif yönde hareket eder. Bu gibi durumlarda, o eksen için cevap -1; aksi takdirde +1 olmalıdır.
- Bu konum, GOHOME yazılımı için sıfır noktasını tanımlamak için son işlemci tarafından kullanılır. Eğer cevap NO ise, FROM yazılımında belirlenen konum sıfır konumu olarak alınabilir.

†Bir sayı

‡Birim M30 verilen cevapla tanımlanır.

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3'ncü Basım, R. ve Brook, N.Y.: IBM, 1985
(Copyright International Business Machines Corporation) IBM Müşahhasuzun Kanlısalatılmıdır (SFE01414-2)



Şekil 12-2 Kesici hareket sınırlarının hazırlanması. C: İzin verilen kesici hareket alanı; D: Genellikle tezgah koordinat sisteminin sıfırının belirlendiği alan.

harekete sahip ise de, birçok NC'li tezgah kesiciyi iki eksenle hareket ettirir.

Koordinat eksenini için sorulan M19 sorusuna "-1" cevabı, Geometrik Bölümün, O eksenindeki koordinat çıktısının işaretini değiştirmesine sebep olur. Birçok durumda, bir APT programındaki koordinat eksen varolan cevapların kullanıldığı tezgah eksenleriyle aynıdır. M20'ye verilen cevapla tanımlanan park konumu GOHOME yazılımını işlemek için kullanılır. M21, M22 ve M23 sorularının cevapları, iş milinin çalışması için ve kesicinin seçimi ile bağlanması için gerekli zamanı hesaplamak için kullanılır.

M25'e verilecek YES cevabı, işleme döngüsü yeteneğine sahip NC'li tezgahlar için kullanılır. Kesici sınır konumlarını kontrol eden tamburların olmadığı modern SD'li tezgahların kullanımından bu yana, M26'dan M29'a kadar sorular gözardı edilmektedir. Bu durumda var olan cevap değerleri kullanılır.

12.2.2 Denetleyici Tanımlama Soruları

NC'li tezgah işlemleri ve hareket, NC kontrolcüsü tarafından kontrol edilir. NC'li tezgahın kontrolcüsünün tasarımı, tezgahın anlayabileceği hareket ve fonksiyonu hazırlar. Bölüm 2'de açıklandığı gibi, farklı tasarımlı NC'li tezgahlarda dört farklı NC blok formatı kullanılır. Her formatta, farklı NC kelimeleri değişik biçimlerde tanımlanır. Bundan dolayı, NC kontrolcü fonksiyonları ve NC kelime formatları, son işlemciye tanımlanmalıdır.

DAPP, kontrolcü karakteristiklerine bağlı olarak 21 soru hazırlar. Bu sorular Tablo 12-7'de listelenmiştir. C06'dan C20'ye kadar olan sorular, son işlemci tarafından delikli şerit veya yazılı çıktıda işlenecek NC kodlarının formatının

hazırlanmasında kullanılır. Genellikle, C06'da sözü edilen geri döndürme kodu, tırnaklı kaset okuyucu kullanıldığında, kasedin geri dönüşünü durdurma için kullanılır. Tırnaksız kaset okuyucu kullanıldığında bu kodun kullanımına gerek yoktur. C07 sorusu, tezgahın kullanıcı veya programlama kılavuzuna göre cevaplanmalıdır. FANUC 6MB kontrolcüsünde programın başlangıcını belirtmek için EOB (en of block - blok sonu) kodu kullanılır.

C08 sorusuna verilen cevap, kontrolcünün kullanacağı ve hazırlayacağı son işlemcinin işleyeceği NC kodlarının miktarını belirler. OUTGRD, 20 x 10 dizisi olduğuna göre, DAPP tarafından ele alınacak kodların en fazla sayısı 20'dir. C09'dan C20'ye kadarki sorular, C08 sorusuna verilen cevaplardaki kod sayısı kadar cevaplanmalıdır. Bu sorulara verilen cevapların herbir seti sadece bir koda ait olup, kodun formatını hazırlar. Soruların verilen cevapların bir koddan diğerine göre ayırt edilmesi için, bir harf (A'dan T'ye kadar) C09'dan C20'ye kadarki soru numaralarına eklenir. Birinci kod için A, ikincisi için B ve 20'nci için T olduğu gibi. Örneğin, C08'e verilen cevabın 12 olduğunu varsayalım. 12, liste sırasına göre son işlemci'deki tanımlanacak N,G,X,Y,Z,I,J,K,M,F,S ve T kodlarını belirtir. Sonra N koduna bağlı bir grup kod, C09A, C10A, ..., C20A, G koduna bağ olarak C09B, C10B, ..., C20B gibi numaralanır. Son olarak T koduna bağlı olarak C09L, C10L, ..., C20L şeklinde numaralama yapılır. C09 ile C20 arasındaki kodlardan geriye kalanlar sadece bir kez cevaplanır. Bu örnekte, blok sıra numarasını tanımlayan N ilk önce tanımlanır, bundan dolayı da C21 sorusuna verilen cevap 1'dir.

Bir NC kodunun çıktı listesindeki konumunu tanımlamak için C19 sorusu cevaplanır. Bundan dolayı, farklı NC kodları için C19 sorularının cevapları, NC bloğundaki NC kodlarının sıralı bir çıktısını hazırlar.

12.2.3 Yazılı Çıktı ve Delikli Şeridin Formatına Bağlı Sorular

Bu iki soru setine verilen cevaplar, delikli şerit ve yazılı çıktının tanımlanmasında gerek duyulan bilgiyi sağlar. Delikli Şerit Hazırlama Soruları (Tablo 12-8), delikli şeridin formatının tanımlanmasında ihtiyaç duyulan cevapları gerektirir. Bu amaçla 4-6 inç uzunluğunda şerit kullanılır. Şerit, okuyucuya doğru olarak yerleştirildiğinde, programın okunması gerçekleştirilir. P2 sorusuna verilen "YES" cevabı, NC şeridinin sürücü kısmını hazırlayan, okunabilir karakterlerde, PARTNO yazılımıyla belirlenen bilgisayar yazılımına sahiptir. Şeridin sürücü kısmında kullanılan karakter normal olarak bir boşluktur. NC'li tezgahın programlama kılavuzu; P3'den P06'ya kadar olan soruların cevaplanmasından önce iyice öğrenilmelidir. Kelime adresi blok formatının kullandığı modern CNC'li tezgahlarda, P06 sorusu değerlendirilmeyebilir.

Düzeltilme Bölümündeki E01 ve E02 sorularına verilen cevaplarla tasarlanmış son işlemcinin ismi ve Takım Tezgahı Modülünün (MTM) program numarası

Tablo 12-7

TABLO 12-7 KONTROLÇÜ TANIMLAMA SORULARI

Soru No. (1'nci nota bakınız)	Soru	Cevap (2'nci nota bakınız)	Varolan değer	Birim	Cevapla ayarlanan değişken
C01	Kontrolcü dairesel enterpolasyon hareketine sahip mi?	YES/NO	NO		CNTOUR (INTEGER*4) = 1 NO 2 YES
C02	Dairesel enterpolasyon fonksiyonu, iptal edilinceye kadar geçerli mi?	YES/NO	NO		DATBUF(37) = 0 NO için 1 veya 2 YES
C03	Dairesel enterpolasyonun en büyük çapı?	r†	99999.99	†	MAXRAD (REAL*8)
C04	Kontrolcü çözümlülüğü veya en düşük artışı koordinatı?	d†	0.00001	†	LIMIT(6)
C05	Bir NC bloğunun oluşumu için en az zaman?	t†	1.0	saniye	TIME(2)
C06	NC bilgisinin ilk bloğundan önce, NC programında ihtiyaç duyulan "Rewind Stop" kodu?	YES/NO	NO		IRSTO (LOGICAL) = TRUE, YES için FALSE, NO için
C07	NC bilgisinin ilk bloğundan önce, NC programında ihtiyaç duyulan "EOB" (blok sonu) kodu?	YES/NO	NO		IEOB (LOGICAL)
C08	Kontrolcü veya son işlemcinin ele alabileceği NC kodumiktari?	n†			= TRUE, YES için FALSE, NO için
C09 ^s	NC kodu harf adresine sahip mi? Eğer cevap YES ise, harf giriniz. Aksi takdirde sıfır yazınız.	harf veya 0	0		OUTGRD(n+1) = 999
C10 ^s	Herbir NC kodu öntünde "tab" kodu var mı? Eğer cevap YES ise sayısını veriniz?	YES/NO n1†	NO		OUTGRD(i,1) = 0 NO için n1 YES ve n1 için
C11 ^s	Koddaki toplam dijital sayısı? Kelime adres formatındaki alfabetik harf ve ondalık noktanın sayılmadığına dikkat ediniz?	n2†	0		OUTGRD(i,3)†

Tablo 12-7

C12 [§]	Koddaki kesirli dijitalerin sayısı?	n_3 [†]	0	OUTGRD(4,4) [‡]
C13 [§]	Yazılacak ondalık nokta?	YES/NO	NO	OUTGRD(4,5) [‡]
C14 [§]	Delinecek ondalık nokta?	YES/NO	NO	OUTGRD(4,6) [‡]
C15 [§]	Yazılmış listede pozitif değerler için + işareti istenir?	YES/NO	NO	OUTGRD(4,5) [‡]
C16 [§]	Delinmiş çıktıda pozitif değerler için + işareti istenir?	YES/NO	NO	OUTGRD(4,6) [‡]
C17 [§]	Bu NC kodunda	YES/NO	NO	OUTGRD(4,7) [‡]
C18 [§]	Bu NC kodundaki	YES/NO	NO	OUTGRD(4,7) [‡]
C19 [§]	Yazılacak kodun kofordaki başlangıcı?	n_4 [†]	1	OUTGRD(4,8) [‡]
C20 [§]	Bu kod için ihtiyaç duyulan delikli şerit çıktısı?	YES/NO	NO	OUTGRD(4,9) [‡]
C21 [§]	C09'dan C20'ye kadar olan sorularla tanımlanan kod listesindeki blok sıra numarasını temsil eden kod numarası nedir (N)?	n_5 [†]	0	DISKID (INTEGER*4)

NOTLAR:

1. Girdi kartının 1-3'ncü kolonlarında tanımlanır.

2. Girdi kartının 5 ve 72'nci kolonları arasında herhangi bir yerde tanımlanır.

[†] Bir sayı[‡] M30 sorusuna verilen cevapla birim tanımlanır.[§] Soru numarasından sonra bir ilave alfabetik karaktere ihtiyaç duyulur.[¶] "i" indisi C09'dan C20'ye kadarki soruların numaralarından sonra, belirli bir karakterle tanımlanmış alfabetik numaraya eşittir. Örnek;

i = 1; C09A,.....,C20A için

i = 2; C09B,.....,C20B için

ve

i = 20; C09T,.....,C20T için

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Nümerik Kontrol İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısıtlıdır (SH201414-2);

Tablo 12-8

TABLO 12-8 DELİKLİ ŞERİT HAZIRLAMA SORULARI

Soru No [†]	Soru	Cevap [‡]	Var olan cevap	Cevapla ayarlanacak değişken
P01	NC delikli şeridin uzunluğu?	n (bir sayı)	0 [§]	LDRLEN (INTEGER*4)
P02	PARTNO'da tanımlanan açıklayıcı bilgi, şeridin başlangıcından önce kişinin anlayacağı formda delinecek mi?	YES/NO	NO	IPART (LOGICAL) = TRUE YES için FALSE NO için
P03	Şeritte kullanılan karakter?	NC'li tezgahın kullanım kılavuzuna bakın	Delik	LCHAR(1) (REAL*8)
P04	"Rewind-stop" kodunu tanımlayan karakter?	NC'li tezgahın kullanım kılavuzuna bakın	%	LCHAR(2) (REAL*8)
P05	"End-of-block" kodunu tanımlayan karakter?	NC'li tezgahın kullanım kılavuzuna bakın	\$	LCHAR(3) (REAL*8)
P06	"tab" kodunu tanımlayan karakter?	NC'li tezgahın kullanım kılavuzuna bakın	*	TABCOD (LOGICAL*1)

NOTLAR:

[†] Girdi kartında 1-3 kolonlarında tanımlanır.[‡] 5 ve 72'nci kolonlar arasında herhangi bir yerde tanımlanır.[§] Ölçüm birimi M30'a verilen cevapla uyumluluk sağlamalıdır.

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısıtlıdır (SH201414-2);

tanımlanır. DAPP, ona kadar son işlemciyi ele alabildiği için, E02'ye verilecek değer, 1-10 arasındaki herhangi biri olabilir. E03 ve E04 sorularına verilen cevaplar, koordinat listesini ve yazılı çıktındaki herbir sayfanın başlığını yazdırmak için kullanılır. Herbir soru için iki girdi kartı (satır) gereklidir. Yazılı girdi 1 ile 70'nci kolonlar arasında listelendiği için, 120 karakterlik bilgi girilebilir. Sonuçta girilen bilgiler çıktı sayfasının başlığı ve koordinat listesinin birinci ile ikinci başlık satırları olur.

12-3 "BLOCK DATA" ALTPROGRAMINI HAZIRLAMAK İÇİN KULLANILAN İŞLEM SIRASI

Bir son işlemcinin tasarımındaki ilk adım, NC'li tezgah kontrol sistemine bağlı gerekli bilgi ve son işlemcinin arzu edilen çıktı formatından meydana gelir. Bu bilgi daha sonra DAPP Soru Listesini cevaplamak için kullanılır. Verilen cevaplar Kısım 12.2'de tanımlanan formata bağlı olarak girilmelidir. DAPP içindeki QUEST isimli program, son işlemci çıktısındaki istekleri ve NC'li tezgaha bağlı DAPP değişkenlerinin ilk değerlerinin atamasını yapan BLOCK DATA programının oluşumunu sağlamak için APT-AC NC işlemcisi içinde yer alır. VM/CMS ve OS/MVS ortamı altındaki BLOCK DATA alt programın oluşumu için programlar NC işlemcisi tarafından sağlanır. Bunlar Ek D'de verilmiştir.

Takip eden konularda, BLOCK DATA programının oluşumu için işlem sırası ve detayları açıklayabilmek için, örnek olarak Cadillac NC-100 torna tezgahı ve FANUC 6T kontrol sistemi ele alınmıştır. Cadillac NC-100, bağlama, silindirik çubuk ve mil işleri için hasarlanmış üniversal bir torna tezgahıdır. Tezgah, pnömatik olarak tahrik edilen, dört kesici konumlu döner başlığa sahiptir. Ayrıca parçaların sağ taraftan desteklenmesi için kullanılabilen, elle kumandalı punta mevcuttur. Bu, elle kumanda edilen tezgahla benzerlik gösterir. Fener mili hareketi, araba, enine araba ve döner başlık NC kontrolcüsü tarafından çalıştırıldığı için, geleneksel tezgahla arasında belirgin farklar vardır. Şekil 22(a)'da tornanın koordinat sistemi ve yapısı şematik olarak verilmiştir. Bu tezgahın kapasitesi aşağıdaki gibidir.

Punta-kızak yüksekliği	23 $\frac{1}{4}$ inç
Enine arabanın hareket miktarı	11 $\frac{13}{32}$ inç
Puntalar arası mesafe	39 $\frac{3}{8}$ inç

İş mili iki hız alanı içinde çalıştırılabilir: Yüksek devir (70-1980 dev/dak) ve düşük devir (12-249 dev/dak). İş mili hızı, % 1 artışlı en büyük hız yüzdesi gibi tanımlanabilir. FANUC 6T kontrol sisteminin fonksiyonları ve özellikleri, Bölüm 2'de tanımlanmıştır.

Tablo 12-9

Soru No†	Soru	Cevap‡	Var olan cevap	Cevapla ayarlanacak değişken
E01	Son işlemcinin ismi?	İlkinin alfabetik olması gereken 1-6 alfa sayısal karakter. (MACHIN yazılımında kullanıldığı gibi).	DAPP	PPCODE (REAL*8)
E02	Takım Tezgahı Modülünün (MTM) program numarası?	1 ile 10 arasında bir sayı	1	PPPARM (REAL*8)
E03§	Yazılacak herbir sayfanın başlık yazısı?	11-70'nci kolonlar arasında tanımlanan alfa sayısal yazı	Boşluk	ULINE0 (15 elemanlı REAL*8 dizisi)
E04§	Koordinat listesinin başlığının birinci satır?	11-70'nci kolonlar arasında tanımlanan alfa sayısal yazı	Boşluk	ULINE1 (15 elemanlı REAL*8 dizisi)
E05§	Koordinat listesinin başlığının ikinci satır?	11-70'nci kolonlar arasında tanımlanan alfa sayısal yazı	Boşluk	ULINE2 (15 elemanlı REAL*8 dizisi)
E06	Listelenen herbir sayfanın sonunda yazılacak işleme zamanı?	YES/NO (Cevap eğer YES ise, 11-70'nci kolonlar arasında yazma zamanını giriniz).	NO	ULINE3 (15 elemanlı REAL*4 dizisi)
E07	Listelenen herbir sayfanın sonunda, yazılacak kağıt şerit uzunluğu? (Uzunluk birimini M30'a verilen cevapla hazırlanır).	(Cevap eğer YES ise, 11-70'nci kolonlar arasında yazma zamanını giriniz).	NO	ULINE4 (15 elemanlı REAL*4 dizisi)

NOTLAR:

† Girdi kartında 1-3'ncü kolonlarda tanımlanır.

‡ E01 ve E02 soruları için 5 ve 72'nci kolonları arasında tanımlanır. E03-E07 için cevaplar 11-70'nci kolonlar arasında cevaplanmalıdır. E03, E04 ve E05 soruları için iki girdi kartı (satır) gereklidir.

§ Bu soru için iki girdi kartı gereklidir. Cevap yazılı çıktının 133 karakter uzunluğundaki başlığı alacaktır.

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal İşlemcisi: Program Kullanımı Kılavuzu, 3'ncü Basım. R'ye Brook, N.Y.: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısıtlanmıştır. (SH201414-2).

TABLO 12-9 DÜZENLEME SORULARI

DAPP Soru Listesine istenen formatta verilen cevapların tam listesi Şekil 12-3'de verilmiştir.

İlk üç soru, M01, M02 ve M03, X, Y ve Z eksenlerinde NC'li tezgahın hareket alanıyla ilgilidir. Kısım 8.1'de işaret edilen belirli bir problem, torna için bu sorulara cevaplandırılırken gözönünde bulundurulmalıdır. Bilindiği gibi torna tezgahları için APT'de programlama yapılırken X-Y düzleminde iki boyutta işlem yapılır.

```

M01 -38.0 +38.0
M02 -12.5 +12.5
M03 0.0 0.0
M04 +600.0
M05 NO
M06 +600.0
M07 1.0
M08 12.0
M09 NO
M15 NO
M16 YES
M17 NO
M18 NO
M19 +1 +1 +1
M20 NO
M21 1.0
M22 2.0
M23 0
M24 H
M25 YES
M30 IN
C01 YES
C02 NO
C03 3937.0
C04 0.0001
C05 1.0
C06 YES
C07 YES
C08 12
C09A N
C09B G
C09C X
C09D Z
C09E I
C09F K
C09G F
C09H S
C09I T
C09J M
C09K U
C09L W
C10A NO
C10B NO
C10C NO
C10D NO
C10E NO
C10F NO

```

Şekil 12-3 Cadillac NC-100 torna/FANUC 6T kontrol sistemi için DAPP soru listesine istenen formatında verilen cevapların listesi.

```

C10G NO
C10H NO
C10I NO
C10J NO
C10K NO
C10L NO
C11A 4
C11B 2
C11C 7
C11D 7
C11E 7
C11F 7
C11G 7
C11H 2
C11I 4
C11J 2
C11K 7
C11L 7
C12A 0
C12B 0
C12C 4
C12D 4
C12E 4
C12F 4
C12G 4
C12H 0
C12I 0
C12J 0
C12K 4
C12L 4
C13A NO
C13B NO
C13C YES
C13D YES
C13E YES
C13F YES
C13G YES
C13H NO
C13I NO
C13J NO
C13K YES
C13L YES
C14A NO
C14B NO
C14C YES
C14D YES
C14E YES
C14F YES
C14G YES
C14H NO
C14I NO
C14J NO
C14K YES
C14L YES
C15A NO
C15B NO
C15C NO
C15D NO
C15E NO
C15F NO
C15G NO
C15H NO

```

Şekil 12-3

C15I NO
 C15J NO
 C15K NO
 C15L NO
 C16A NO
 C16B NO
 C16C NO
 C16D NO
 C16E NO
 C16F NO
 C16G NO
 C16H NO
 C16I NO
 C16J NO
 C16K NO
 C16L NO
 C17A NO
 C17B NO
 C17C YES
 C17D YES
 C17E YES
 C17F YES
 C17G YES
 C17H NO
 C17I NO
 C17J NO
 C17K YES
 C17L YES
 C18A NO
 C18B NO
 C18C NO
 C18D NO
 C18E NO
 C18F NO
 C18G NO
 C18H NO
 C18I NO
 C18J NO
 C18K NO
 C18L NO
 C19A 1
 C19B 7
 C19C 11
 C19D 20
 C19E 30
 C19F 39
 C19G 67
 C19H 75
 C19I 80
 C19J 86
 C19K 48
 C19L 57
 C20A YES
 C20B YES
 C20C YES
 C20D YES
 C20E YES
 C20F YES
 C20G YES
 C20H YES
 C20I YES
 C20J YES

Şekil 12-3

```

C20K YES
C20L YES
C21 1
P01 6
P02 NO
P03
P04 8
P05 *
P06
E01 LATH2
E02 08
E03 NC-100 LATH/FANUC-6T CONTROLLER SYSTEM
E03 POSTPROCESSOR
E04 N G X Z I K U W
E04 F S T M TIME ISN
E05
E06 YES TOTAL MACHINING TIME (IN HR-MIN-SEC) =
E07 YES TOTAL TAPE LENGTH (IN FEET) =
  
```

Şekil 12-3

NC'li bir tornada X-Z düzlemindeki kesici hareketi, APT koordinat sisteminin X-Y düzlemindeki programlanmış harekete sahiptir. Son işlemci ile işlenen CLDATA, APT programından çıkarıldığında, APT programındaki X ve Y eksenlerini, tezgahın Z ve -X eksenlerine dönüştürür (Şekil 8-1).

Bir APT programındaki kesici yolu X-Y düzleminin üst yarısında tanımlanır (bu durumda Y koordinatlarının çoğu pozitifdir). Bu, arzu edilen kesici yolunun tezgah Z eksenine bağlı olarak simetrik görünüşüdür. Eğer son işlemci APT'deki +x ve +y koordinatlarını sırasıyla NC programındaki +z ve +x koordinatlarına dönüştürürse, NC programında tanımlanan kesici yolu hedeflenen kesici yoluyla aynıdır. Bununla beraber, bir APT programının yazımında kullanılacak doğru kesicinin ve iş mili dönüş yönünün seçiminde gerekli özen gösterilmelidir. TLLFT ve TLRGT yazılımlarının kullanımı ile APT programlama süresince kesici uç yarıçapı da gözönünde bulundurulmalıdır. Program sonucunda CLDATA, kesici uç yarıçapı merkezinin koordinatlarının bir bütünüdür. Bundan dolayı, kesici uç yarıçapının gözönünde tutulması, CLFILE'daki kesici yolunun NC kodlarına dönüşümü süresince gereksizdir.

Bölüm 8.1'de işaret edilen diğer yaklaşım seçilirse, APT programlama süresince kesici yolu X-Y düzleminin alt yarısında tasarlanır. Bu durumda, APT'deki y koordinatların çoğu negatif olduğundan, APT'deki x ve y koordinatlarını tezgah koordinat sisteminde z ve -x koordinatlarına dönüştürmek gerekir.

Programlamanın çok daha kolay olmasından dolayı ilk yaklaşım seçilmiştir. M01 ve M02'ye verilen cevaplar Z ve X eksenlerindeki hareket alanıdır. Herbir eksenindeki hareketin gerçek sınırları, tezgah kapasite parametrelerinden çıkarılır. Koordinat sisteminin merkezi bu alan içinde herhangi bir yere ayarlandığında,

hareketin üst ve alt sınırları da aynı olacak şekilde ayar edilir. APT koordinat sisteminin Z eksenini kullanılmadığından cevap sıfırdır. M19'a verilecek cevap, her üç eksen içinde +1'dir.

Döner başlığın 90° dönüşü için gerekli zaman 2 saniye civarındadır. Bundan dolayı M22'ye verilecek cevap 2.0'dır. Bu torna tezgahında kesici magazini olmadığı için, M23'e verilen cevap "0"dir. Diğer sorulara verilecek cevaplar açık olduğu için, daha fazla açıklamaya gerek duyulmamıştır.

NC kontrolcusuna bağlı sorular, tezgahın kullanım kılavuzuna ve NC kontrolcusuna bağlı olarak cevaplanır. Verilecek cevaplar açık ve nettir. Bununla beraber, CO2 sorusu, sadece kontrolcu tarafından kontrol edilmeyen fonksiyonlarla ilgilidir. Bu soruya verilen cevap "NO" ise, DATBUF(37)'nin sıfır olduğu veya APT içinde dairesel enterpolasyona gerek olmadığına anlamına gelir (ARCSIN/OFF gibi). Cevap "YES" olduğunda, DATBUF(37) 1 veya 2 değerliğini alır. Bu değerlendirme de, programda dairesel enterpolasyonu istediği anlamına gelir. Bu yüzden, cevap normal olarak "NO" olmalıdır. CO3'e verilen cevap, FANUC 6T kontrol sisteminde programlanabilir en büyük yarıçap olan 3937.0 inç'tir. Geriye kalan sorular, Bölüm 2'de detaylı olarak tanımlanan, FANUC 6T sisteminin programlama isteklerine bağlı olarak cevaplandırılır.

Delikli şerit hazırlama sorularına bağlı olarak, bu tezgah için NC şeridi, delik karakterli 6 inçlik şerit gerektirir. Şeridin geriye döndürmesini durdurma kodu "99" ve blok sonu kodu "*"dır. Herbir NC kodundan önce herhangi bir tabulatör komuduna gerek yoktur.

Son işlemcinin ismi LATH2 ve TTM'nin numarası 8'dir. E01 ve E02 soruları bunlara bağlı olarak cevaplanmalıdır. E03'den E07'ye kadar olan sorular, yazılı çıktıda aranan özelliklere bağlı olarak cevaplanmalıdır. Okuyucu bu cevapları, Bölüm 15'de verilen son işlemci çıktı listesi ile karşılaştırmalıdır.

DAPP soru bankasına verilecek cevaplar bir FORTRAN dosyası şeklinde saklanmalıdır. Örneğin, VM/CMS ortamında, dosya aşağıdaki isim, tip ve modda saklanabilir:

ANSWER FORTRAN A1

Aşağıda verilen iş oluşturma programı QUESTEXE¹¹ (VM/CMS ortamında) cevapları işlemek için daha sonra kullanılabilir. Bu program FORTRAN BLOCK DATA alt programı tarafından oluşturulur (Şekil 12-4).

&CONTROL ALL

FILEDEF FT06F001 PRINTER

FILEDEF FT08F001 DISK F1 F1 A4 (XTENT 50 RECFM F BLKSITE 400)

FILEDEF FT07F001 PUNCH

```

BLOCK DATA
C*****
C
C   INITIALIZATION OF DAPP COMMON BLOCK QUES08.
C
C*****
      IMPLICIT      REAL*8 (A-H,O-Z)
      COMMON      /QUES08/
A      COORIS (3) ,DATBUF (85) ,HOME (3) ,LIMITS (11) ,MAXRAD
B      ,PPCODE ,PPARM ,TPLIMT ,ULINE0 (15) ,ULINE1 (15)
C      ,ULINE2 (15) ,XCONVT ,YCONVT ,ZCONVT ,XTRA (2)
D      ,TIME (8) ,ULINE3 (15) ,ULINE4 (15)
E      ,CLDRMS ,CNTOUR ,DISKID ,DPDRMS ,DRUMZE
F      ,IGOHM ,IGOTAB ,IRPMOD ,IZHOR ,LDRLN
G      ,LEADSW ,NLINE ,OUTGRD (20,10) ,R10 ,STRISQ
H      ,SIMOSW (4) ,TABLGO ,ZMOD (2) ,Z10
I      ,CLKPRT ,EOB ,IEOB ,IPART ,IRSTO
J      ,LCHAR (2) ,TABCOD ,TAPPRT ,TAPMET

      REAL*8      LIMITS ,MAXRAD
      REAL*4      TIME ,ULINE3 ,ULINE4
      INTEGER*4   CLDRMS ,CNTOUR ,DISKID ,DPDRMS
A      ,DRUMZE ,OUTGRD ,R10 ,STRISQ ,SIMOSW
B      ,TABLGO ,ZMOD ,Z10
      LOGICAL*1   CLKPRT ,EOB ,IEOB ,IPART
A      ,IRSTO ,LCHAR ,TABCOD ,TAPPRT ,TAPMET
      EQUIVALENCE (ZDEPTH,DATBUF (29)) (ZFINAL,DATEBUF (30))
      REAL*8      CONVEN (3)
      EQUIVALENCE (CONVEN (1) ,XCDIG ,XCONVT) , (YCDIG ,YCONVT) , (ZCDIG ,ZCONVT)
      INTEGER      ASSIGN (2) ,CYCTYP (2)
      EQUIVALENCE (ASSIGN (1) ,DATBUF (20)) , (CYCTYP (1) ,DATEBUF (28))
      LOGICAL*1   LOGCAL (6) ,OUTSUX (40)
      EQUIVALENCE (LOGCAL (1) ,DATEBUF (31)) , (OUTSUX (1) ,OUTGRD (1,1))
      DATA COORIS / 0.0 D0, 0.0 D0, 0.0 D0/
      DATA DATBUF / 8H99999.99, 8H99999.99, 8H99999.99,
A 0.0 D0, 0.0 D0, 0.0 D0,
B 0.0 D0, 0.0 D0, 1.0000D0, 600.0000D0,
C 0.0 D0, 8H99999.99, 12.0000D0, 8H99999.99,
D 8H99999.99, 8H99999.99, 8H99999.99, 8H99999.99,
E 8H99999.99, 0.0 D0, 8H99999.99, 8H99999.99,
F 0.0 D0, 8H99999.99, 8H99999.99, 8H99999.99,
G 8H99999.99, 8H99999.99, 8H99999.99, 0.0 D0,
H 8H99999.99, 8H99999.99, 8H99999.99, 8H99999.99,
I 8H99999.99, 0.0 D0, 48*8H99999.99/
      DATA HOME / 0.0 D0, 0.0 D0, 0.0 D0/
      DATA LIMITS / 38.0000D0, 12.5000D0, 0.0 D0,
A 8H99999.99, 8H99999.99, 0.000100D0, 8H99999.99,
B 600.0000D0, -38.0000D0, -12.5000D0, 0.0 D0/
      DATA MAXRAD / 3937.0000D0/
      DATA PPCODE /6H TURN/, PPARM / 8.0D0/
      DATA TPLIMT /36000000.0D0/
      DATA ULINE0 / 8HNC-100 L,8HATH/FANU,8HC-6T CON,8HROLLER
A,8HSYSTEM ,8H ,8H ,8H POST,8HPROCESSO,8HR
B,8H ,8H ,8H ,8H /
      DATA ULINE1 / 8HN G ,8H X ,8H Z ,8H I
A,8H K ,8H U,8H W,8H ,8H F ,8H S T
B,8H M ,8H ,8H ,8H TIME,8H ISN/,
      DATA ULINE2 / 8H ,8H ,8H ,8H
A,8H ,8H ,8H ,8H ,8H
B,8H ,8H ,8H ,8H /
      DATA ULINE3 / 4HTOTA,4HL MA,4HCHIN,4HING ,4HTIME,4H(IN ,4HHR-M

```

Şekil 12-4 Şekil 12-3'de verilen cevaplardan oluşturulmuş BLOCK DATA programı.


```

A      ,4HIN-S,4HEC) ,4H= ,4H ,4H ,4H ,4H ,4H ,4H /
DATA  ULINE4 / 4H ,4H ,4H TOT,4HAL T,4HAPE ,4HLENG,4HITH(I
A      ,4HN FE,4HET) ,4H= ,4H ,4H ,4H ,4H ,4H ,4H /
DATA  XCONVT / 1.0D0/, XCONVT / 1.0D0/, ZCONVT / 1.0D0/
DATA  XTRA / 1.0000D0, 1.0000D0/
DATA  TIME / 0.0 , 1.0000, 1.0000, 1.0000,
A      2.0000, 0.0 , 0.0 , 0.0 /
DATA  CLDRMS / 0/, CNTOUR / 2/, DISKID / 1/, DFORMS / 0/,
A      DRUMZE / 0/, IGOHOM / 0/, IGOTAB / 0/, IRPMOD / 0/,
B      IZHOR / 1/, LDRLN / 6/, LEADSW / 0/, NLINE / 46/
DATA  OUTGRD / 1HN, 1HG, 1HX, 1HZ, 1HI, 1HK, 1HF, 1HS, 1HT, 1HM,
A      1HU, 1HW, 1HO, 1HO, 1HO, 1HO, 1HO, 1HO, 1HO, 1HO,
B      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
C      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 4, 4, 4,
D      4, 0, 0, 0, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0,
E      0, 4, 2, 7, 7, 7, 7, 2, 4, 2, 7, 7, 7,
F      0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 2,
G      2, 2, 1, 1, 1, 2, 2, 0, 0, 0, 0, 0, 0,
H      0, 0, 1, 1, 2, 2, 2, 2, 1, 1, 1, 2,
I      2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2,
J      2, 2, 2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0,
K      0, 0, 0, 1, 7, 11, 20, 30, 39, 67, 75, 80, 86,
L      48, 57, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
M      1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
N      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
O      0, 0, 0, 999, 0, 0, 0, 0, 0, 0, 0, 0, 0,
DATA  R10 / 0/, STRISQ / 1/, SIMOSW / 1, 0, 0, 0/,
A      TABLGO / 0/, ZMOD / 0, 1/, Z10 / 0/,
DATA  CLKPRT /T/, EOB /1H*/ , IEOB /T/, IPART /F/,
A      IRSTO /T/, LCHAR /1H ,1H*/ , TABCOD /1H*/ ,
B      TAPPRT /T/, TAPMET /F/
END

```

Şekil 12-4

```

FILEDEF FT05F001 DISK ANSWER FORTRAN
LOADMOD QUEST
START

```

OS/MVS ortamı için İş Kontrol Dili (JCL) (Job-Control-Language) yazılımı EK D'de verilmiştir. Bu programdaki DATA yazılımlarının Şekil 12-3'de verilen cevaplarla yapılan karşılaştırmasından, bunların uygunluğu ortaya çıkar.

PROBLEMLER

12.1 FANUC 6MB kontrolcülü NC'li düşey freze tezgahına ve özelliklerine bağlı olarak verilenlerle, DAPP soru listesini cevaplayınız ve BLOCK DATA programını oluşturunuz.

a. Tezgah özellikleri:

- (1) Tezgah tablasının programlanabilir hareket değeri
X yönünde 27.5 inç
Y yönünde 17.75 inç
- (2) İş milinin programlanabilir hareket değeri
Z yönünde 4.0 inç
Devir sayısı 10-3000 dev/dak
- (3) X, Y ve Z eksenlerinde programlanabilir doğrusal ve noktadan noktaya hareket.
- (4) Üç koordinat düzlemindeki programlanabilir dairesel enterpolasyon hareketi.
- (5) Programlanabilir çevrim hareketi (Bölüm 2'de tanımlandığı gibi)
- (6) Hızlı çapraz ilerleme miktarı (programlanamaz) 480 inç/dak.
- (7) Programlanabilir talaş kaldırma ilerlemesi 0.01-150 inç/dak.
- (8) Elle kesici değişimi

b. Kontrolcu özellikleri

- (1) Kullanılan NC kodları: N,G,X,Y,Z,R,F,S,M,Q,P (zaman)
- (2) NC kelimelerinin formatları; Tablo 2-3'e bakınız
- (3) Delikli şerit uzunluğu: 6 inç
- (4) Şeritte kullanılan karakter: Delik
- (5) "end-of-block" kodunu tanımlayan karakter; ","
- (6) "rewind-stop" kodunu tanımlayan karakter; "%"

Son işlemci ismi SAMPLE ve tezgah bağımlı işlem yordamının program numarası 8'dir.

12.2 DAPP soru listesinin cevaplanmasında ihtiyaç duyulan, son işlemci çıktısında aranan özellikler ve NC'li tezgah kontrol sisteminin karakteristiklerinin yanısıra, DAPP tabanlı bir son işlemcinin tasarımında ihtiyaç duyulan diğer bilgi çeşitleri nelerdir?

Bölüm 13

DAPP Tabanlı Son İşlemcinin Tezgah-Bağımsız İşleme Yordamları

Daha önce işaret edildiği gibi, tezgah-bağımsız işleme yordamları genellikle her son işlemci için aynıdır ve DAPP içinde teşkil edilmiştir. Bir son işlemci tasarımcısının, DAPP tabanlı son işlemci tasarımı süresince bu yordamlara dikkat etmesi gerekmez. Bununla beraber, tezgah bağımlı işlem yordamının yazılabilmesi, MTM ve son işlemci tasarımının yapılabilmesi için bu yordamların anlaşılması gerekir.

Bu bölümde tartışılan, tezgah bağımsız işlem yordamları aşağıdaki kısımlara ayrılabilir:

1. İlk değer atama bölümü
2. Düzenleme bölümü
3. Okuyucu bölümü
4. Genel işleme bölümü
5. Çıktı bölümü

Herbir bölüm için, bir yordamın fonksiyonu, yordamın oluşundan sonuçlanan değişkenlerdeki değişimler ve kontrol akışı açıklanacaktır.

13.1 İLK DEĞER ATAMA BÖLÜMÜ

Bu bölümde, ilk değerler DAPP COMMON yazılımlarında belirlenen değişkenlere tahsis edilir. Tezgahla ilgili olanlar; BLOCK DATA programındaki DATA

yazılımları ile (diğer bir deyişle, DAPP soru listesine verilen cevaplarla hazırlanır) veya kullanıcı tarafından hazırlanan TTM alt yordamları ile ilk değerlendirilirler. DAPP'da, soru listesine bağlı olarak verilen cevaplarla ilk değerleri atanmış değişkenler, QUEST isimli COMMON yazılımında tanımlanır. Bölüm 12'de verilen metodun kullanımı ile oluşturulan BLOCK DATA programında, COMMON yazılımı QUESNn olarak isimlenir. Kelimenin son iki hanesi "nn" ilgili TTM'nin program numarasını belirler. Bu bölümdeki LDCOM isimli özel yordam, BLOCK DATA program içindeki QUESNn isimli COMMON bloğunu, DAPP'ın QUEST isimli COMMON bloğuna taşır. Böylelikle değişkenlerin ilk değer ataması yapılmış olur.

DAPP içinde tezgahla ilgili olan diğer değişkenler COMMON yazılımlarında, DBFPRT ve SEGCOM isimleriyle tanımlanır. Listelenen dosyada yazılacak DATBUF elemanlarının alanı, DBFPRT isimli COMMON yazılımındaki değişkenler ile tanımlanır.

```
COMMON/DBFPRT/N1,ISTART(5),LENGTH(5)
```

Bu yazılımda;

N1 = Yazılacak DATBUF yerleşim alıcılarının sayısını

ISTART(i) = i nci alanın başlangıç konumu gibi bir DATBUF elemanını tanımlayan numarayı (i = 1, ..., 5)

LENGTH(i) = i nci alan da, yazılacak (i = 1, ..., 5), DATBUF elemanlarının toplam sayısını tanımlayan numarayı

ifade eder. Bu değişkenler DLWDBF yordamının parametreleridir. Bu yordam, tezgah bağımsız bölüm tarafından işlenen DATBUF dizisinin tanımlanmış elemanlarını yazdırır. ISTART ve LENGTH dizilerinin boyutu 5 olduğunda, en büyük alan değeri 5 olmalıdır. Bu değişkenlerin aşağıdaki gibi ilk değer almaları sağlanır:

$$N1 = 1$$

$$LENGTH(i) = \begin{cases} 85 & (i = 1) \\ 1 & (i = 2, \dots, 5) \end{cases}$$

$$ISTART(j) = 1 \quad (j = 1, \dots, 5)$$

Bu değerler 85 DATBUF elemanının tümünün bir alan gibi gözönüne alınacağı ve yazılacağını belirtir. Bu değişkenler ve ortak kayıt alanı (DBFPRT isimli) TTM içinde tanımlanmalıdır. Sonra yazım yordamı DLWDBF, belirlenmiş DATBUF elemanlarının yazımı için çağırılır. Örneğin, eğer aşağıdaki yazılımlar TTM'de tanımlanmışsa, 1'den 3'e ve 36'dan 71'e kadar olan DATBUF dizisi, isim ve değerleri

```
COMMON/DBFPRT/N1,ISTART(5), LENGTH(5)
```

```
.....
.....
N1 = 2
ISTART(1) = 1
ISTART(2) = 36
LENGTH(1) = 3
LENGTH(2) = 36
```

ile birlikte yazılır.*

SEGCOR isimli COMMON yazılımında tanımlanan değişkenler, kesicinin lunu iki veya daha fazla kısma bölmek için kullanılır. Ayrıca bu değişkenlerin değeri alması bu bölümde sağlanır ve eğer ihtiyaç duyulursa TTM'de tekrar tanımlanır. Bu değişkenlerin açıklaması Bölüm 14'de verilmiştir.

Tezgaah bağımsız değişkenleri DAPP'ın iç değişkenleridir. Değerlerinin ataması DAPP tarafından yapılır ve genellikle gerekmedikçe değiştirilmez. Bu değişkenlerin bir listesi IBM'in kullanım kılavuzunda⁷ açıklamalarıyla beraber bulunabilir.

Bir CLFILE'in işlenmesi süresince, ilk değer atama işlemi başlangıçta bir defaya mahsus olmak üzere yapılır. Sonra kontrol Düzenleme Bölümüne gönderilir.

13.2 DÜZENLEME BÖLÜMÜ

DATBUF dizisi içindeki bir kısım bilgi; bir set CLDATA'nın işlenmesinde ve çıktısının alınmasından sonra, yeni bir set CLDATA'nın okunmasından önce güncelleştirilmeli veya sıfıra ayarlanmalıdır (ANULL)[†]. Bu düzenleme işlemi CLDATA'nın doğru işlenmesi için gereklidir. Örneğin, bir set CLDATA işlenip çıktısı alındıktan sonra, hedef noktasının X, Y ve Z koordinatlarını tanımlayan DATBUF (1), (2) ve (3) ANULL'a ayarlanmalıdır. Bunun anlamı, hedef noktası olarak herhangi bir girdinin kayıt edilmediğidir. Eğer, sonraki CLDATA setinin kesici konumu sadece X ekseninde değişirse, bu eksene bağlı DATBUF konumları DATBUF(1) Okuyucu Bölümü tarafından güncelleştirilir ve diğer iki konumun değerleri, DATBUF(2) ve (3) sabit tutulur (örnek ANULL). Böylece kesicinin bu iki eksendeki hareketi oluşturulmaz. Bu üç konum, Düzenleme Bölümü tarafından sıfıra ayarlanmaz ve kesicinin Y ve Z eksenlerindeki yanlış hareketi meydana getirilmez. DATBUF dizisinin 85 elemanının çalışması, Düzenleme Bölümünün ayarlanmayan elemanların aşağıdaki gibi açığa çıkmasını sağlar:

DATBUF(i) i = 10,11,13-17,20,23,26-30,34,35

* Bölüm 15'deki örneğe bak.

† Tablo 12-1'deki 5'nci dipnota bak.

Kontrol Düzenleme Bölümüne girdiğinde, gagalama delme döngüsünün aktif olup olmadığını kontrolü için, LOGCAL (3) değerinin testi yapılır. Eğer LOGCAL(3) değeri doğru (TRUE) ise, diğer bir deyişle birkaç delme döngüsü oluşturulmuş ise, DATBUF dizisi güncelleştirilmemelidir. Bunun sonucu, kontrol delme döngüsünün diğer adımını işlemek için DLWCYC paket-çevrim yordamına geçer. Aksi takdirde, düzenleme yordamı işleme konur.

Tablo 13-1'de, düzenleme yordamının işleme konmasından sonra, çeşitli DATBUF konumlarının değerleri listelenmiştir.

Düzenleme Bölümü iki durumda çağrılır:

1. CLFILE'in işlenmesinin başlangıcında, ilk değer atama işlemi yapıldıktan sonra, Düzenleme Bölümü çalıştırılır.

TABLO 13-1 DÜZENLEME BÖLÜMÜ TARAFINDAN TEKRAR AYARLANAN DATBUF ELEMANLARININ DEĞERLERİ

i	Düzenleme Bölümünün İcrasından Sonra DATBUF(i) Değeri
1,2,3	ANULL
4	DATBUF(4) + DATBUF (7)
5	DATBUF(5) + DATBUF (8)
6	DATBUF(6) + DATBUF (9)
7,8,9	0.0
10,11	*
12	0.0 Eğer M05 sorusuna verilen cevap NO ise * Eğer M05 sorusuna verilen cevap YES ise
13-17	*
18,19	ANULL
20	*
21,22	0.0
23	*
24,	0.0
25	ANULL
26-30	*
31	LOGCAL(j) = FALSE j = 1,2,4
32-33	ANULL
34,35	*
36	Eğer tezgaah dairesel enterpolasyon fonksiyonuna sahipse ANULL olarak ayarlanır, aksi takdirde bu eleman tekrar ayarlanmaz.
37	*
38-41	DATBUF(36)'nın aynısı
42-71	ANULL
72-85	*

*Bu eleman(lar) Düzenleme Bölümü tarafından yeniden ayarlanmaz. 464'ncü referanstan alınmıştır. (Referans)

Otomatik Programlanmış Takımlar - Geliştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3üncü Basım. R'ye Brook, N.Y: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2):

† Kısım 13.4.2.3'e bak.

2. Bir set CLDATA kaydının işlenmesinin tamamlanmasında, kontrol DAPP ana işlemi tarafından Düzenleme Bölümüne gönderilir.

Her iki durumda da, DATBUF dizisi Tablo 13-1'deki gibi güncelleştirilir ve kontrol sonra Okuyucu Bölümüne gönderilir.

13.3 OKUYUCU BÖLÜMÜ

Bir APT programındaki bilgi, CLREAD isimli yordamı içeren Okuyucu Bölümü CLFILE'dan DAPP'a geçirilir. Bir set CLDATA kaydı CLREAD tarafından okunduğunda, kayıttaki bilgi aşağıda listelenen parametrelere transfer edilir:

- IRECNO (INTEGER) = O anki CLDATA setinin kayıt numarası
 INNBUF = 252 konumlu, o anki bir setlik CLDATA'dan okunan bilginin saklanması için geçici bellek alanı gibi kullanılan REAL*8 dizisi
 IWORDS (INTEGER) = CLREAD yordamından INNBUF dizisine transfer edilen kelime sayısı
 CONTRL (INTEGER) = O anki CLDATA setinin kayıt sınıfı (Tablo 13-2'ye bakınız)
 NSUBCL (INTEGER) = O anki CLDATA setini daha fazla tanımlayan parametre (farklı kayıt sınıfları için, bu parametre farklı yollarda tanımlanır⁷).

Örneğin, CLDATA'nın 11'nci kaydının; (girdi kartının 73-80'nci kolonlarındaki) CLPRNT, son işlemci komutu olduğunu varsayalım. Bölüm 11'de belirtildiği gibi bu yazılımdan iki kayıt, 1000 ve 2000 sınıfı, oluşturulur. 1000 kayıt sınıfı için CLREAD yordamı tarafından seçilen parametreler aşağıdaki gibidir:

- IRECNO = 11
 INNBUF(1) = 00000020 (INNBUF(i), i = 2,..., veya 252, kayıt içermez)
 IWORDS = 1
 CONTRL = 6 (1000 kayıt sınıfının, sınıf numarası 6'dır. Tablo 132'ye bakınız)
 WSUBCL = İç sıra numarası (ISN)

2000 kayıt sınıfı (son işlemci komutu) aşağıdaki parametrelere transfer edilir:

- IRECNO = 11
 INNBUF(i) = Son işlemci kelimesi CLPRNT'den sonra herhangi bir parametre

metre yoksa, kayıt içermez (ikincil kelime ve parametrelili son işlemci kelimeleri için, bu dizi, esas kelimedenden sonra tanımlanmış parametre ve ikincil kelimeleri içerir) (i = 1'den 252'ye kadar).

- IWORDS = 0
 CONTRL = 1 (Parametresiz bir son işlemci kelimesinin sınıf numarası 1'dir. Tablo 13-2'ye bakınız)
 NSUBCL = Bu son işlemci kelimesinin CLSETP dizisi⁷ içindeki konum numarası. (Bu konum numarası DAPP tarafından işlenecek esas son işlemci kelimeleri içermek üzere DAPP içinde ayarlanır).

TABLO 13-2 DAPP TARAFINDAN İŞLENEN BİLGİ TİPLERİ

Kayıt Sınıfı	CONTRL	Kayıdın Tanımlaması
1000	6	Okunan CLDATA kaydının iç sıra numarasını (ISN)
2000	1	Parametre(ler)siz ve/veya ikincil kelime(ler)siz son işlemci kelimesi
	2	Parametre(ler)li ve/veya ikincil kelime(ler)li son işlemci kelimesi
3000	7	Dairesel sürücü yüzey
5000	3	Kesici eksenini daima Z ekseninde (MULTAX/OFF) olduğunda, FROM, GOOLTA veya GOTO yazılımıyla tanımlanan kesici konum(lar)u
	5	Yukarıdaki MULTAX/ON'lu hali
9000	9	Kesici eksenini uyum bilgisi (MULTAX/ON veya OFF)
	10	Ölçü birimi bilgisi
13000	8	Hata göstergesi
14000	4	CLFILE'in sonunu belirten FINI kaydı

5000 kayıt sınıfı için, NC işlemcisi tarafından işlenen ve oluşturulan kesici konumları numarası 1'den büyük olabilir. Böylece INNBUF dizisi bu noktaların X, Y ve Z koordinatlarını içerir.

Bir son işlemci komudunun parametrelerine transferinden önce, komut içindeki esas son işlemci kelimesi, DAPP'ın izin verdiği esas son işlemci kelimelerini içeren CLSETP dizisi ile karşılaştırılır. İkincil son işlemci kelimeleri, DAPP tarafından kabul edilen ikincil son işlemci kelimelerini içeren dizi ile karşılaştırılır. Eğer bir son işlemci kelimesi bu dizilerden birinde bulunursa, dizi içindeki konumu, esas kelime için NSUBCL şeklinde saklanır. İkincil kelime(ler) ve parametre(ler) INNBUF(i) şeklinde korunur. Aksi takdirde, CLREAD yordamı tarafından gözard edilirdi.

On çeşit bilgi DAPP tarafından ele alınabilir, bunlar TABLO 132'de verilmiştir. Okuyucu, farklı kayıt tipleri için CLREAD yordamı ile transfer edilen parametrelerin detayları hakkında IBM'in Kullanım Kılavuzuna⁷ müracaat edebilir.

Bir CLDATA kaydı bir kez okunduğunda, CONTRL parametresi ile işaret edilen kayıt sınıfına uygun Genel İşleme Bölümündeki işleme yordamına gönderilir. Bilgi daha sonra kodlanır, işlenir ve uygun dizi (DATBUF gibi) veya değişkenlerde saklanır.

13.4 GENEL İŞLEME BÖLÜMÜ

Genel İşleme Bölümü, CLDATA'nın tezgah bağımsız işlemini harekete geçirir. Genel fonksiyonlarından biri, kesici konum bilgisini nakletmektir. CLDATA'da verilen kesici konumları parça koordinat sistemine bağlı olarak tanımlanır. Kesicinin diğerine göre konum farklılığı (offset), NC işlemcisi ile bir APT programının işlenmesi süresince gözönünde bulundurulmaz. Bununla beraber, son işlemci çıktısındaki (parçanın NC programı) kesici konumları tezgah koordinat sistemine göre tanımlanmalıdır. Kesicinin konum farkına göre doğrultma, kesici değişimlerinin olduğu durumlarda gereklidir. Bundan dolayı, Genel İşleme Bölümü, CLDATA'dan kesici konum koordinatlarını, tezgah koordinat sistemi ve kesici konum farklılığına bağlı olarak transfer edebilmelidir.

Genel İşleme Bölümünün diğer temel fonksiyonu; son işlemci komutlarının (kelimeleri) kodlarını açmak, kolayca geri çağrılabilir ve tezgah-bağımlı işleme bölümü (MTM) tarafından kullanılmasını sağlamak amacıyla, dönüştürülmüş bilgiyi uygun değişkenler şeklinde saklamaktır.

Genel İşleme Bölümü 10 ayrı alt bölüme ayrılabilir. Bunlar DAPP içinde, Tablo 13-2'de verilen 10 çeşit kayıt tipini işleyen kod sürücüsü, ayırıcı alanlar olarak isimlendirilir. Kesici konum bilgisi ve son işlemci dışındaki kayıtlar, uygun kod ayırıcı alanlarda işlenir ve sonra ya tezgah-bağımlı işlem bölümüne, MTM veya doğrudan DAPP Çıktı Bölümüne gönderilir. Son işlemci kelimelerinin işlenmesi veya kesici-konum bilgisi, Tablo 13-3'de verilen alt yordamların yardımıyla işleme sokulur.

Genel İşleme Bölümü tarafından işlenen bilginin açıklanmasına örnek olarak aşağıdakiler verilmiştir:

13.4.1 Yardımcı Bölüm

Bu bölüm altı işlem yordamından meydana gelir. Bunlardan dördü, DLWPP0, DLWUP1, DLWPP2 ve DLWUP2 parametrelili veya parametresiz son işlemci kelimelerini ele alır. İki hesaplama yordamı, DLWROT ve DLWORG, sırasıyla döner tablanın konumunu tanımlayan ROTABL ve parça koordinat sistemine göre tezgah koordinat sisteminin merkez noktasını tanımlayan ORIGIN yazılımını işlemek üzere hazırlanır. ROTABL ve ORIGIN aynı zamanda son işlemci komutlarıdır (kelimeler).

Bir son işlemci komudu, APT-AC NC işlemcisi ile işlenmeden CLFILE geçirilen bir kelimedir. Sonra son işlemci tarafından işlenir. DAPP'ın Yardımcı Bölümü tarafından işlenen son işlemci kelime veya komutları sınırlıdır. Bununla beraber, birçok uygulama için yeterlidirler.

TABLO 13-3 GENEL İŞLEME BÖLÜMÜNDEKİ ALT YORDAMLAR

Bölüm	Alt yordam	Fonksiyon
Yardımcı	DLWPP1	Parametresiz son işlemci kelimelerini işleme
	DLWPP2	Parametrelili son işlemci kelimelerini işleme
	DLWUP1	DLWPP1 tarafından ele alınamayan parametresiz son işlemci kelimelerini işleme
	DLWUP2	DLWPP2 tarafından ele alınamayan parametrelili son işlemci kelimelerini işleme
	DLWORG	Son işlemci ORIGIN komutunu işleme
Geometrik	DLWROT	Son işlemci ROTABL komutunu işleme
	DLWLIN	Doğrusal enterpolasyon hareketini işleme
	DLWCIR	Dairesel enterpolasyon hareketini işleme
	DLWCYC	Talaş kaldırma döngü hareketini işleme

13.4.1.1 DLWPP1, DLWUP2, DLWPP2 ve DLWUP2 Yordamları

Tablo 13-4'de DLWPP1 yordamı tarafından işlenebilen parametresiz son işlemci kelimeleri ve işleme sonuçları verilmiştir. Bu kelimeler, APT programlamasında da sık sık kullanılır. DLWPP1 tarafından işlenemeyen parametresiz son işlemci kelimeleri, standart dışı olanlardır. Bunlar DLWPP1 tarafından DLWUP1'e transfer edilir. Yine de, bu kelimelerin DLWUP1 tarafından alınmasından sonra işleme yapılmayıp sadece 6117 numara ile hata mesajı verilir. Sonra bu kelime gözardı edilir ve kontrol DLWPP1'e döner.

Tablo 13-4'de görüldüğü gibi, DLWPP1 ve DLWUP1 yordamları parametresiz bir son işlemci kelimesinin işlemini tamamlandıktan sonra, kontrol şu üç bölümden birine geçer: Okuyucu Bölümü, Geometrik Bölüm veya MTM.

Parametrelili son işlemci kelimeleri, DLWPP2 ve DLWUP2 tarafından işlenir. DLWPP2 tarafından işlenen ve Tablo 13-5'de verilen 24 kelime vardır. DLWPP2 tarafından işlenemeyenler DLWUP2'ye gönderilir. DLWUP2'ye gönderilen işlenemeyen fonksiyondan sonra hata mesajı 6118 verilir ve kontrol tekrar DLWPP2'ye geri döner.

DLWPP2 ve DLWUP2'nin bir son işlemci kelimesinin işlemini bitirdikten sonra, kontrol başka bir CLDATA kaydının okunması için istenen kelimeye bağlı olarak Okuyucu Bölümüne veya Çıktı Bölümüne gider (Tablo 13-5).

DLWUP1 ve DLWUP2 yordamları, DAPP'ın ilave son işlemci kelimelerini kabul etme imkanını verir. Bunlar standart dışı olup, farklı üretimler için farklı işleme yordamlarına ihtiyaç gösterir. Bundan dolayı, bunlar için işleme yordamı çatısı sadece DLWUP1 ve DLWUP2 tarafından sağlanır. Eğer kullanıcı, bu ilave

TABLO 13-4 DLWPP1 TARAFINDAN İŞLENEN PARAMETRESİZ SON İŞLEMÇİ KELİMELEERİ

Son işlemci kelimesi	Anlamı	Bu kayıdın işlem sonucu
END	Bu yazılım, parça programının bitişini tanımlamak için APT programının sonunda kullanılır. Eğer kullanılmamış ise, FINI okunduğunda, DAPP otomatik olarak oluşur. Son işlemci tarafından oluşturulan uygun NC kodu M02 veya M30'dur.	1. DATBUF(25) = 9001 2. Bu kayıt işlendikten sonra, kontrol MTM'ye gider.
OPSTOP	Bu komut, NC programı içinde isteğe bağlı M01 durumunda kodunun oluşumu veya tanımlanması için kullanılır.	1. DATBUF(25) = 9002 2. Bu kayıdın işlenmesinden sonra kontrol MTM'ye gider.
STOP	Bu komut NC programına ara vermek için kullanılan M00 kodunun durumu için kullanılır.	1. DATBUF(25) = 9003 2. Kayıdın işlenmesinden sonra kontrol MTM'ye gider.
RAPID	G00 hızlı ilerleme kodunun oluşumunu sağlamak için kullanılır. M05 sorusuna verilen cevap N0 olursa bir sonraki hareket için veya M05 sorusunun cevabı YES olduğunda ilerleme yazılımının iptal edildiğinin gösterimi için etkili bir komuttur.	1. DATBUF(12) = 1 2. Kayıdın işlenmesinden sonra kontrol MTM'ye geçer.
GOHOME	Kesicinin, M20 sorusuna verilen cevapla veya FROM yazılımında M20 sorusuna verilen N0 cevabıyla, hızlı hareket yaparak park konumuna gitmesi için kullanılır.	1. a. DATBUF(1), (2) ve (3), sırasıyla park konumunun x, y ve z koordinatları veya FROM konumundaki değerler. b. ATBUF(12) = 1 c. M14 sorusuna verilen carbin YES olmasıyla, ilave bir yazılım, döner tablaya belirlenen "0" konumuna gitmesini işaret eder. 2. Kontrol, Okuyucu Bölümü ile doğrusal enterpolasyon hareketi işlem rutinine,

TABLO 13-4 DLWPP1 TARAFINDAN İŞLENEN PARAMETRESİZ SON İŞLEMÇİ KELİMELEERİ (devam)

Son işlemci kelimesi	Anlamı	Bu kayıdın işlem sonucu
GOCLER	Bu komut, döngü hareketinde CLEARP komuduyla tanımlanan kesici yüzeyine kesicinin boşluk mesafesine CLEARP/... yazılımı ile hızlı ilerleme hareketini kullanarak (?) çıkılmasından sonra tanımlanmalıdır. Detaylı bilgi için (Kısım 13.4.2.3) DLWCYC rutinine bakınız.	1. a. DATBUF(1), (2) ve (3), CLEARP/... yazılımıyla tanımlanan boşluk düzlemine bağlı olarak ayarlanmalıdır. b. DATBUF(12) = 1 2. CLEARP/... yazılımı tanımlanmamış ise 6112 hata mesajı verilir. 3. Herhangi bir hata oluşmaz ise, kontrol Okuyucu Bölümünde DLWLN rutinine gider. 1. a. DATBUF(3) tanımlanmış boşluk düzlemine bağlı olarak ayarlanır. b. DATBUF(12) = 1 2. CLRSRF yazılımı tanımlanmamış ise 6113 hata komutu verilir. 3. Hata oluşması durumunda, kontrol DLWLN rutinine gider. Bu yazılım nerede yazılırsa yazılsın, önemli bir hata olması durumunda bile, Çıktı Bölümünün delikli şeridin hazırlanması konumuna gelmesi sağlanır. Bu kayıdın işlenmesinden sonra, kontrol tekrar Okuyucu Bölümüne geri döner.
RETRCT	CLRSRF/... son işlemci komutundan sonra tanımlanmalıdır. CLRSRF ile X-Y düzlemine paralel boşluk düzlemi tanımlanır. Komut ile kesicinin boşluk düzlemine hareketi termin edilir. Detaylı bilgi için Kısım 13.4.2.3'e bakınız.	
RESET	Bu yazılım, son işlem süresince veya CLDATA'da önemli bir hata olsa bile, DAPP'1 SD delikli kartın oluşumuna yöneltir.	

Otomatik Programlanmış Takımlar - Geliştirilmiştir Çevresel Sayısal Denetim İşlemcisi: Programı Kullanım Kılavuzu, 3ncü Basım. R'ye Brook, N.Y.: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2).

TABLO 13-5 DLWPP2 YORDAMI İLE İŞLENEN PARAMETRELERİ SON İŞLEMÇİ KOMUTLARI

Son işlemci komutu	Anlamı	Bu komutun işlem sonucu
ARCSLP/ $\begin{Bmatrix} \text{ON} \\ \text{OFF} \end{Bmatrix}$	Kesici konum bilgisini işlemek için DLWCIR dairesel enterpolasyon rutininin kullanımını kontrol etmek için kullanılır. ON seçildiğinde, DLWCIR çağılır. Aksı takdirde DLWLN doğrusal enterpolasyon tünini kullanılır. \square	1. DATBUF(37)= 1 ON için 0 OFF için 2. Eğer yazılım hatası varsa, 6001 hata mesajı verilir. 3. Bu komut işlendikten sonra, 5000 sınıfını okumak için, kontrol Okuyucu Bölümüne geri döner, GOTO/(çok nokta) bilgisi 1. DATBUF(25) = n 2. Yazılım hatası varsa, 6002 hata mesajı verilir. 3. Bu yazılım işlendikten sonra, kontrol TTM'ye gider. 1. Boşluk (açıklık) düzlemi, doğru veya nokta programca ayarlanır ve saklanır. 2. Yazılım hatası durumunda 6003 hata mesajı verilir. 3. Bu yazılımın işlemesinden sonra kontrol Okuyucu Bölümüne döner.
AUXFUN/n	Bu komut, SD programında M kodunu oluşturmak için kullanılır.	
#: skaler büyüklük		
CLEARP/ $\begin{Bmatrix} \text{XYPLAN} \\ \text{YZPLAN} \\ \text{ZXPLAN} \end{Bmatrix}, d1$	Boşluk düzlemini, doğru veya nokta tanımlaması için kullanılır. Bu yazılım GOCLER ile birlikte, işleme döngüsündeki geri çekme hareketinin tanımlaması için kullanılır. Detaylı bilgi için Kısım 13.4.2.3'e bakınız.	
$\begin{Bmatrix} \text{XYPLAN} \\ \text{YZPLAN} \\ \text{ZXPLAN} \end{Bmatrix}, d2$		
$\begin{Bmatrix} \text{XYPLAN} \\ \text{YZPLAN} \\ \text{ZXPLAN} \end{Bmatrix}, d3$		
d1:	Tanımlanan boşluk düzlemi ile paralel koordinat düzlemleri arasındaki mesafe.	
d1,d2:	Tanımlanan boşluk düzlemi ile paralel koordinat düzlemleri arasındaki mesafeler (tanımlanan iki koordinat düzleminin aynı olmamasına dikkat edilmelidir)	
d1,d2,d3:	Tanımlanan boşluk noktası ve koordinat düzlemlerinin mesafesi (Bu üç koordinat düzlemlerinden herhangi ikisinin aynı olmamasına dikkat edilir).	

TABLO 13-5 DLWPP2 YORDAMI İLE İŞLENEN PARAMETRELERİ SON İŞLEMÇİ KOMUTLARI (devam)

Son işlemci komutu	Anlamı	Bu komutun işlem sonucu
CLRSRF/d	X-Y düzleminde paralel boşluk düzlemini tanımlamak için kullanılır. Bu yazılım, işleme döngüsündeki \square geri çekme hareketini tanımlamak için RETRACT yazılımıyla birlikte kullanılmalıdır.	1. X-Y düzleminde paralel boşluk düzlemi programca ayarlanır ve saklanır. 2. Yazılım hatası durumunda 6021 hata mesajı verilir. 3. Kontrol Okuyucu Bölümüne geri döner.
COOLNT/ $\begin{Bmatrix} \text{FLOO} \\ \text{MIST} \\ \text{TAPKU} \\ \text{ON} \\ \text{OFF} \end{Bmatrix}$	Soğutma sıvısının seçimi ve açılmasının tanımlanması için kullanılır.	1. DATBUF(17) ayarlanır (Tablo 12-1'e bakınız) 2. Yazılım hatası durumunda 6004 hata mesajı verilir. 3. Kontrol Okuyucu Bölümüne döner.
CUTCOM/ $\begin{Bmatrix} \text{LEFT} \\ \text{RIGHT} \\ \text{OFF} \\ \text{#} \end{Bmatrix}$	Tanımlanan X-Y, Y-Z veya Z-X düzleminde kesici yarıçapının ayar tarafının tanımlanması için kullanılır. \dagger	1. DATBUF(27) ayarlanır (Tablo 12-1'e bakınız). 2. Yazılım hatası durumunda 6005 hata mesajı verilir. 3. Kontrol Okuyucu Bölümüne döner.
#: Pozitif skalar büyüklük		
CYCLE/...	Delme, kılavuz çekme veya frezeleme döngüsünün tanımlanması için ihtiyaç duyulan parametreleri belirlemek amacıyla kullanılır. Bu yazılımdaki parametreler, döngü tipini, boşluk yüksekliğini, işleme derinliğini, ilerleme miktarını ve bekleme zamanını içerir.	1. a. DATBUF(31) veya LOGCAL dizisi, tanımlanan döngü tipine bağlı olarak ayarlanır (Kısım 12.1.3 ve 13.4.23'e bakınız) b. YCTYP(2) veya DATBUF(28) tanımlanan döngü tipine göre ayarlanır. c. Boşluk düzleminin, delme ve frezeleme işleminin z koordinatı program tarafından saklanır. d. DATBUF(19) tanımlanan bekleme zamanına göre ayarlanır. 2. Yazılım hatası durumunda 6008 hata mesajı verilir. 3. Kontrol Okuyucu Bölümüne döner.
Bu yazılımın formatı için Kısım 13.4.2.3'deki DLWCYC rutinine bakınız.		

DELAY/I

Kesicinin o anki konumunda t saniye cinsinden beklemesini sağlar.

FEDRAT/ {IPM₁/I
IPR₂/I, MAXIPM₃/I}

f_1, f_2 ve f_3 sıfırdan daha büyük olmalıdır.

IPM: Dakikada inç

IPR: Kesicinin veya iş milinin her dönüşünün inç değeri

MAXIPM: Dakikadaki inç cinsinden en büyük ilerleme miktarı

INSERT...

.....: NC yazılımı

LEADER/n

n= Pozitif skaler büyüklük

LOADTL/I, SETOOL d, e, f

refer için, Kısım 6.6'ya bakınız

Sonraki yazılımlarda tanımlanan kesici hareketi için ilerleme hareketinin tamamında kullanılır.
"MAXIPM, f_1 , f_2 ", M06 sorusuna verilen cevapla f_{max} , en büyük ilerleme miktarının değişiminin f_3 'e yazımı için tanımlanır.

Tanımlanmış çık için veya NC program çıktısındaki NC yazılımı için son işleminin hareketi geçirilmesi için kullanılır.

M30'a verilen cevapla tanımlanan birimlerle, delikli şeridin uzunluğunu tanımlamak için kullanılır.
Bu yazılım, cevabı, P01 sorusunun üzerine yazar.

X, Y ve Z yönlerinde sırasıyla, d, e ve f offset değerlerine sahip t numaralı kesicinin işleme konulması için kullanılır. Bu değerler, sifir (referans) noktasından kartere bağlı kesicinin uç yarıçap merkezinin (torna kesicisi için) veya parmak freze ekseninin değerleridir.

Tablo 13-5

- a. $DATBUF(19) = t$
b. $DATBUF(25) = 9001$
c. $TIME(1) = TIME(1) + t$
- Yazılım hatası durumunda 6006 hata mesajı verilir.
- Kontrol TTM'ye gider.

- a. $DATBUF(12) = 0$
b. $DATBUF(10) = f_1$
 $f_2 - n$

IPM, f_1 için
IPM, f_2 için
(n = Devir sayısı)

- $DATBUF(10), f_1$ veya f_2 izin verilen en büyük ilerleme miktarından, f_{max} , daha büyük olduğu zamanlar da f_{max} 'a ayarlanır.
2. Eğer bir yazılım hatası varsa veya f_1 veya f_2 'nin f_{max} 'dan daha büyük olduğu durumlarda 6107 hata mesajı verilir
- Kontrol Okuyucu Bölümüne geri döner.

Test emeksinin doğrudan transfer eder. Sonra kontrol delikli şeridin veya yazılı çıktının alınması için Çıktı Bölümüne gider.

- Skaler bilgi program tarafından saklanır ve Çıktı Bölümü NC'li delikli şeridi hazırlarken kullanılır.

- Yazılım hatası durumunda 6009 hata mesajı verilir.

- Kontrol Okuyucu Bölümüne döner.

- a. $DATBUF(15) = t$

- d, e ve f değerleri program tarafından saklanır ve kesici konumunun transformasyonu için Geometrik Bölüme naktedilir.

TABLO 13-5 DLWPP2 YORDAMI İLE İŞLENEN PARAMETRELERİ SON İŞLEMCİ KOMUTLARI (devam)

Son işlemci komudu	Anlamı	Bu komutun işlem sonucu
MACHIN/ $ppname, n1, OPTION, n2, n3, n4$	CLDATA'nın işlenmesi için kullanılacak (MTM ve BLOCK DATA programı gibi) son işleminin tanımlanması amacıyla kullanılır. Ayrıca bilgiyi son işlemeiye geçirir.	2. Yazılım hatası durumunda 6020 hata mesajı verilir. 3. Kontrol MTM'ye gider. 1. DAPP, itiyavaş duyulduğunda pname ve n1 parametreleri ile tanımlanan BLOCK DATA ve MTM programını çalıştırır. $n2$ 'ye göre uygun çıktı rutini seçilir. Ayrıca aşağıdaki değişkenler ayarlanır: DATBUF(34) = $n3$ DATBUF(35) = $n4$
Parametrelerin anlamları için Kısım 6'ya bakınız.		2. Yazılım hatası durumunda 6010 hata mesajı verilir. 3. Bu yazılımın işlenmesinden sonra kontrol Okuyucu Bölümüne döner.
MCHTOL/ {ON OFF HIGH LOW n}	Köşelerde kesicinin/işin yavaşlaması ve hızlanması için MTM tarafından istenen işlem toleransını tanımlamak için kullanılır. (Detaylı bilgi için Kısım 14.3.3.1, Bölüm 14'e bakınız).	1. a. $DATBUF(25) = 9005$ b. $DATBUF(26) = 1$ ON için 2 OFF için 3 HIGH için 4 COW için n için
n= Skaler büyüklük		2. Yazılım hatası durumunda 6011 hata mesajı verilir. 3. Bu yazılımdan sonra, kontrol MTM'ye döner.
OPSKIP/ {ON OFF}	OPSKIP/ON yazılımı, OPSKIP/OFF'la iptal edilinceye kadar, son işleminin herbir NC yazılımının başına "/" işaretini koymasını sağlar. NC yazılımındaki "/"'la tezgah durdurma tuşuna basıldığında o yazılımın NC kontrolcüsü tarafından göz ardı edilmesini temin eder.	1. Uygun iş parametre, herbir NC yazılımının başına "/" işaretini yerleştirmek için Çıktı Bölümüne komuta etmek amacıyla ayarlanır. 2. Yazılım hatası durumunda 6012 hata mesajı verilir. 3. Kontrol Okuyucu Bölümüne geri döner.

Tablo 13-5

ORIGIN/x,y,z,[a]	Kullanımı ve işleme sonucu için Kısım 13.4.1.3'e bakınız.
PARTNO	Tanımlanan açıklama yazısı, parça işleme programının tanıtımı için kullanılır.
Formatı için Kısım 6.12.1'e bakınız.	
PLABEL/n1,[n2[...[n6]]]]	Şeritte kişinin anlayabileceği n1, ..., n6 kodlarının tanımlanmasını sağlar. Bu kodlar iki karakterli olarak aşağıdakileri ifade eder: TN: (tape No-Şerit numarası) PN: (part No-Parça numarası) OP: (operator No-Kullanıcı numarası) EC: (engineering change-Mühendis değişikliği) MC: (master change-Master değişikliği) MG: (machining group-Tezgaah grubu)
n1, ..., n6 : 1-6 dijiti içeren pozitif skalar büyüklük	
PPRINT/...	Son işlemcinin, yazıyı çıktı listesine eklenmesini temin eder.
....: Açıklama yazısı (En fazla 66 karakter)	
PREFUN/n	SD çıktı programında G kodlarını tanımlamak için kullanılır.
n: Pozitif skalar büyüklük	
ROTABL/[INCR, [ATANGL], a, [CLW], [CCLW], [ROTREF]	(Kullanımı ve işlem sonucu için Kısım 13.4.1.2'ye bakınız)
SELECTL/n	Kesici mağazinde "n" numaralı kesicinin, otomatik kesici değiştiricidaki konumundan hareket ettirilmesi için kullanılır.
n: Pozitif skalar büyüklük	

TABLO 13-5 DLWPP2 YORDAMI İLE İŞLENEN PARAMETRELERİ SON İŞLEMCI KOMUTLARI (devam)

Son işlemci komutu	Anlamı	Bu komutun işlem sonucu
SEQNO/n1,[INCR,n2,n3]	O anki blok için sıra (n1) numarasını tanımlanması Sıra numarası, n1'nci sıra numaralı blokta sonra her n3 blok için n2 kadar artırılır.	1. n1 numarası, n2 ve n3 parametreleri ile ayarlanan durumlara göre güncelleştirilen DATBUF(21) elemanında öncelikte ayarlanır. 2. Yazılım hatası durumunda 6018 hata mesajı verilir. 3. Kontrol Okuyucu Bölümüne döner.
1. n1,n2,n3: Pozitif tamsayı 2. Eğer tanımlanmış ise n2 = n3 = 1'dir 3. Bu yazılım tanımlanmış ise, yazılım		
SEQNO/1		
kabul edilir.		
SPINDL/ ON OFF S	İş milinin dönüş hızı ile yönünün tanımlanması ve milin çalıştırılıp durdurulması için kullanılır.	1. DATBUF(13) ve (14) (Tablo 12-1'e bakınız) ayarlanır. Eğer "s" parametresi tanımlanmış ise, SPINDL yazılımında tanımlanan veya programda tanımlanmış ise M08 sorusuna verilen cevap iş mili devri olarak kullanılır. 2. Yazılım hatası durumunda 6019 hata mesajı verilir. 3. Kontrol Okuyucu Bölümüne döner.

*Aslında, CL-FILE'daki kesici konumları, kesici yançapı için telafi edilir. Bundan dolayı, kesici yançapı telafisinin yönünün (sol veya sağ) tanımlanması amacıyla bu yazılımın kullanılmasına gerek yoktur. Bununla beraber, yazılım, dairesel enterpolasyon düzlemine tanımlanmış veya yazılımdaki "n" skalar ile tanımlanan son işlemciye kullanışlı bilgilerin transferi için kullanılır.
Otomatik Programlanış Takımlar - Getiştirilmiş Çevresel Sayısal Denetim İşlemcisi: Program Kullanım Kılavuzu, 3üncü Basım. R'ye Brook, N.Y.: IBM, 1985 (Courtesy of International Business Machines Corporation). IBM kılavuzundan kısaltılmıştır (SH201414-2):

son işlemci kelimelerinden bir veya bir kaçının işlenmesine ihtiyaç duyarsa, uygun işleme rutinlerinin eklenmesi ile uygun DLWUP1 ve/veya DLWUP2'yi değiştirebilir. Daha detaylı bilgi için IBM'in kullanım kılavuzuna⁷ başvurulabilir.

13.4.1.2 ROTABL Yazılımı ve Yazılımın DLWROT İşleme Yordamı

ROTABL komutu, döner tablanın bir dönüşünü tanımlar. Yazılım iki farklı formata sahiptir. Bunlardan birincisi;

ROTABL/ {INCR
ATANGL} , a, {CLW
CCLW} 'dir.

Yazılımdaki semboller aşağıdakileri ifade eder:

- a = Döner tablanın dönüş açısı
- INCR = O anki konumdan dönüşün tanımlanmış açısının artış değeri
- ATANGL = Döner tablanın, açısal sıfır değerinden mutlak olarak tanımlanmış açı değeri
- CLW = Saat ibresi yönünde ölçülen açı değeri
- CCLW = Saat ibresinin ters yönünde ölçülen açı değeri

Bu yazılım, X, Y ve Z yönlerindeki tezgah hareketi gibi değerlendirilen döner tabla dönüşünü basitçe tanımlamak için kullanılır. Bu yazılımın parça veya tezgah koordinat sistemi üzerinde herhangi bir etkisi yoktur.

ROTABL yazılımının diğer bir formatı, parça koordinat sisteminin merkezini dönüşünü döner tabla ile birlikte tanımlamak için kullanılır. Format aşağıdaki gibidir:

ROTABL/ {INCR
ATANGL} , a, {CLW
CCLW} 'ROTREF

ROTREF ikincil kelimesi, döner tabla ile dönecek koordinat sisteminin merkezini ifade eder. Ayrıca "INCR,a," için döner tabla merkezi etrafında "a" açısı veya "ATANGL,a," için açısal konumu, a, işaret eder. Bunlar CLW veya CCLW kelimeleri ile tanımlanan yönlerde belirlenir.

DAPP'da, döner tablanın ekseninin Y eksenine ve dönüşten sonra değişen yeni parça koordinat sisteminin uzaysal uyumuna paralel olduğu varsayılır. Döner tablanın eksenini Y'den farklı bir yön ise, COPY veya TRACUT yazılımı, keski yolunun dönüşümü için kullanılmalıdır.

Şekil 13-1'de, iki yazılımın anlam farklılıkları şematik olarak gösterilmiştir. Şekil 13-1(a)'da, B yüzeyindeki bir delik, A yüzeyi işlendikten sonra delinecektir.

B yüzeyinin istenen konumuna getirilmesi için, döner tablanın 45° döndürülmesi gerekir (Şekil 13-1(b)). Bu durumda, parça koordinat sistemi değişmeksizin sabit kalır. ROTABL yazılımının ROTREF kelimesiyle kullanımının etkisi Şekil 13-1(c)'de gösterilmiştir.

ROTABL/INCR,90,CCLW,ROTREF

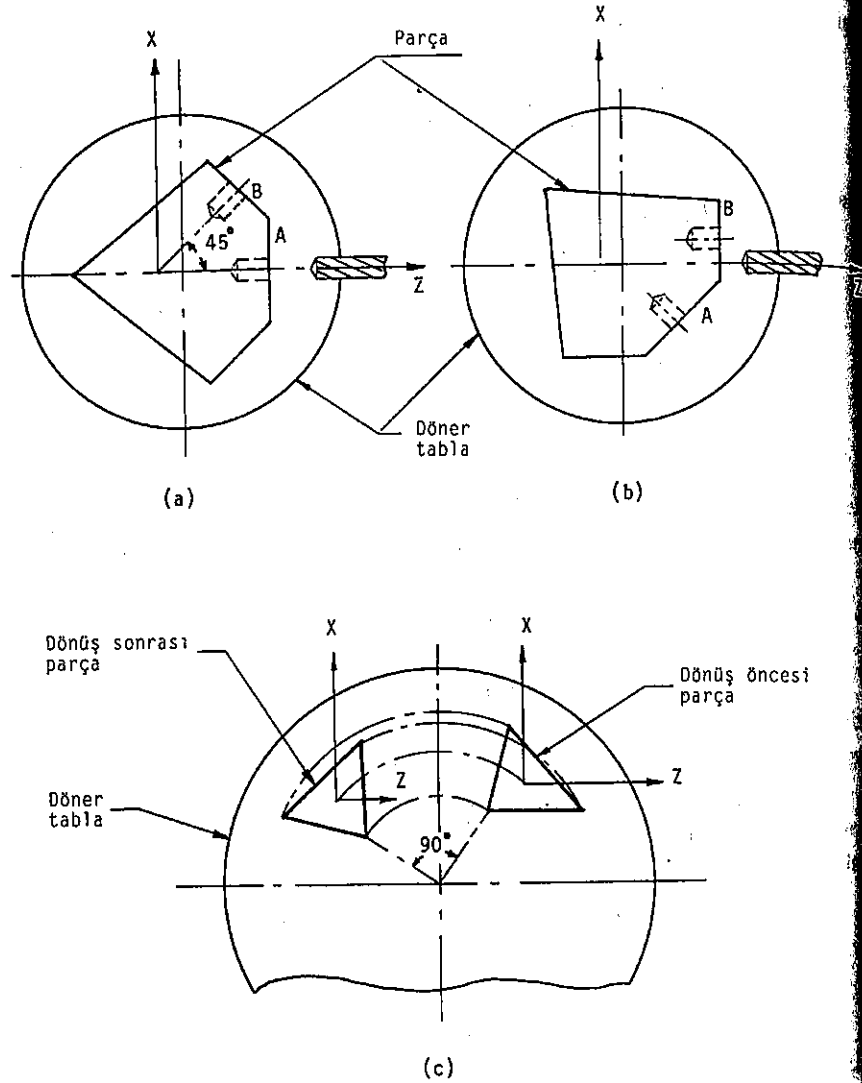
yazılımının icrasından sonra, hem parça hem de parça koordinat sisteminin merkezi 90° döndürülmekle beraber, koordinat eksenlerinin uzaysal konumları değişmeksizin kalır.

ROTABL yazılımı, DLWROT yordamı ile işlenir. İşlem, tasarlanmış sıralama ile oluşturulur. ROTABL yazılımı okunduğunda, parametrelili son işlemci kelimesini işleyen kod çözücü alanına gönderilir. DLWROT yordamı (Şekil 13-2), tanımlanan ROTABL formatını kontrol eder ve herhangi bir yazılım hatası mevcut ise 6016 hata mesajı oluşturulur. Daha sonra kontrol Okuyucu Bölümüne gönderilir. Aksi takdirde, işlem başlar. DATBUF(22), tanımlanan dönüş yönüne göre ayarlanır. Eğer ROTREF okunursa, işlem koordinat sistemi merkezinin X, Y ve Z yerleşimlerini hesaplar. Aksi takdirde; kontrol, zamanı hesaplayan ve döner tabla dönüş yönünün programlanmış konumuna ulaşip-ulaşmadığını kontrol edecek sonraki bölüme geçer. Sonra DATBUF(18), DATBUF(25), ve TIME(1) değişkenleri yukarıda açıklanan hesaplamalara göre ayarlanır. Sonuçta, tabla dönüşünün kesici konum koordinatlarının toplam doğrultması, tezgah koordinat sisteminde döner tabla konumu (M13 sorusuna verilen cevapla tanımlanan) ve kesici offsetleri (farkları) (LOADTL yazılımıyla tanımlanan) hesaplanır ve program tarafından saklanır. Bununla, DAPP'ın her kesici konumunun doğru koordinatlarını hesaplaması sağlanır. Sonuç olarak, işlemden sonra kontrol MTM'ye gider.

13.4.1.3 DLWORG Yordamı

DLWORG yordamı, ORIGIN/x,y,z[a] yazılımını işlemek için kullanılır. Bu yazılımla (1) parça koordinat sistemine göre tezgah koordinat sisteminin merkezi, (2) Z eksenine göre tablanın açısal sıfır konumu tanımlanır. Eğer tezgah bir döner tablaya sahip değilse, "a" parametresine ihtiyaç duyulmaz. Bu durumda, yazılım Bölüm 6.11'de tanımlanan anlama sahip olur (Yazılımın, tezgah koordinat sisteminin merkezini parça koordinat sistemi içinde tanımlandığı gibi).

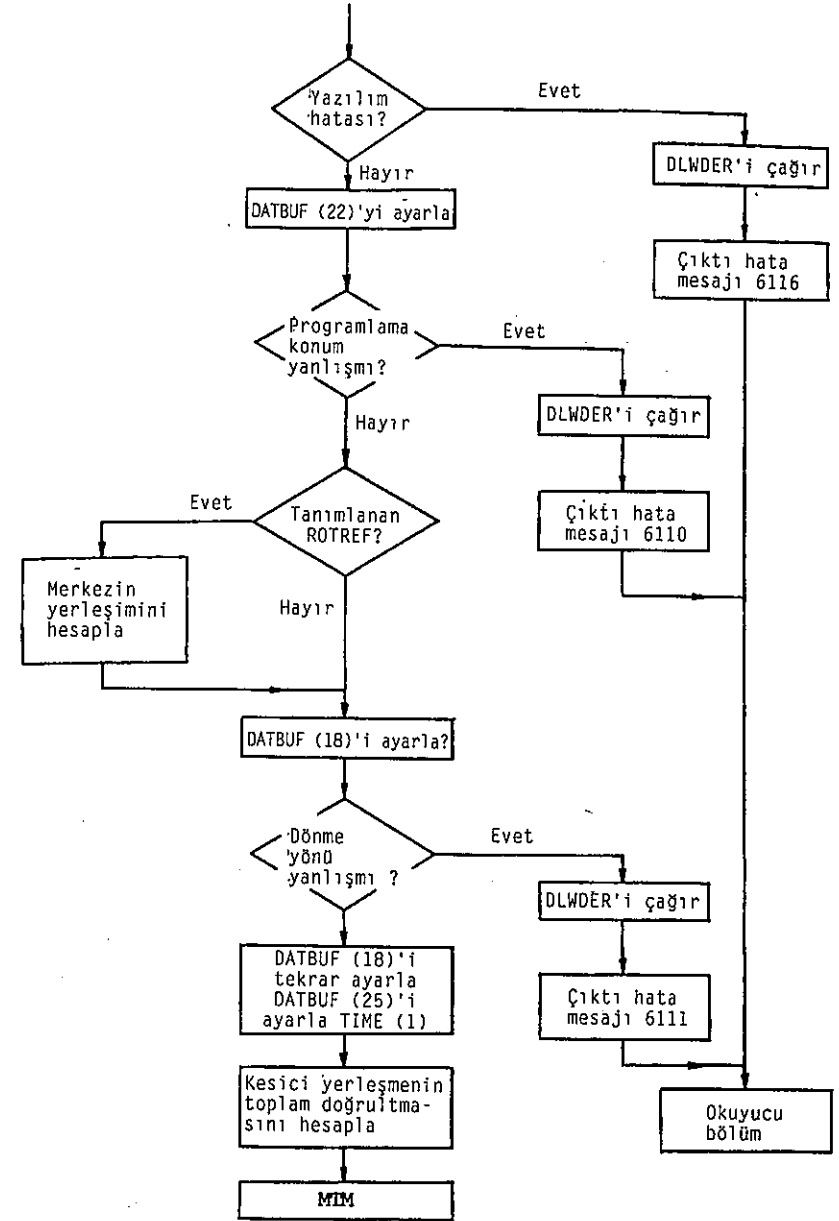
Bir NC'li tezgahda döner tabla var ise, dönme ekseninin Y eksenine paralel olduğu kabul edilir. Buna göre, bu yazılım, parça koordinat sistemi içinde, merkezi döner tablanın merkezinde kabul edilen tezgah koordinat sistemini tanımlar (Şekil 13-3[a]). x, y ve z parametreleri tezgah koordinat sisteminin koordinatlarını veya parça koordinat sistemi içindeki döner tablanın merkezini tanımlar. "a" parametresi,



Şekil 13-1 ROTABL/INCR,45,CLW yazılımının icrasından önce (a) ve sonra (b) parça ve parça koordinat sistemi. (c) Parça koordinat sistemi Z-X'in, döner tabla ile birlikte, merkezinin dönüşü ROTABL/INCR,90,CCLW,ROTREF yazılımıyla tanımlanır.

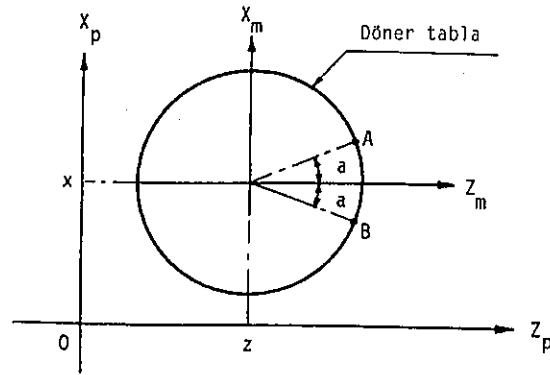
döner tablanın açılma sıfır konumunu tanımlayan açıdır, ve M10 sorusuna verilen cevapla tanımlanan yöndeki +Z ekseninden ölçülür.

Eğer döner tablanın merkezi, tezgah koordinat sisteminin merkezinde değilse, bu yazılım parça koordinat sistemine bağlı döner tabla merkeziyle ilgilidir (Şekil

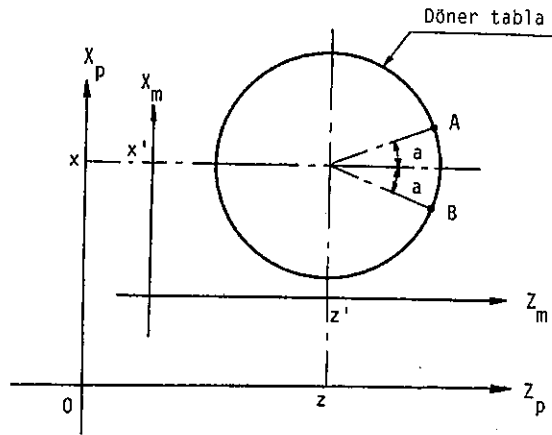


Şekil 13-2 DLWROT işleminin akış şeması. Kontrol, parametresiz son işlemci kelimelerinin işlendiği kod çözücü alandan girilir.

13-3[b]). Bundan dolayı, M13 sorusuna verilen cevaplar, kesici bilgisinin daha ileri dönüşümü için kullanılmalıdır.



(a)



(b)

Şekil 13-3 ORIGIN yazılımının uygulaması. Tezgah koordinat sisteminin merkezi Z_m-X_m ve döner tablanın merkezi (a) ve (b) aynı noktalar değildir.

NOT: Z_p-X_p - parça koordinat sistemi
A - Sadece CCLW yönünde döndürüldüğünde zaman ORIGIN yazılımıyla tanımlanan döner tablanın açısız konumu B - Sadece CLW veya her iki yönde döndürüldüğünde ORIGIN yazılımıyla tanımlanan döner tablanın sıfır konumu.

Bu yordamda işlem oldukça basittir. Öncelikle yazılım kontrol edilir, herhangi bir hata durumunda, 6013 numaralı hata mesajı verilir. Aksi takdirde, parça koordinat sistemine bağlı döner tablanın konumu, polar koordinatlarda hesaplanır. DLWROT yordamının kullanımı için saklanır. Koordinat sisteminin değişimi ile gözlenen kesici konumunun X, Y ve Z yönlerindeki doğrultmaları $-x$, $-y$ ve $-z$ olup (Kısım 6.11'e bakınız), X, Y ve Z yönlerindeki toplam doğrultmalar aşağıdaki gibi hesaplanır:

$$k_x = p_x \cdot (-x + x' + x'')$$

$$k_y = p_y \cdot (-y + y' + y'')$$

$$k_z = p_z \cdot (-z + z' + z'')$$

- x, y, z = ORIGIN yazılımındaki parametreler
 x', y', z' = Tezgah koordinat sistemindeki döner tablanın koordinatları (M13 sorusuna verilen cevaplar gibi)
 x'', y'', z'' = X, Y ve Z yönlerindeki kesici farkları için telafiler
 p_x, p_y, p_z = M19 sorusuna verilen cevaplara bağlı olarak +1 veya -1

Bu değerler, tezgah koordinat sistemine bağlı olarak kesici konumunu hesaplamak için, DAPP'ın Geometrik Bölümünün kullanılması amacıyla program tarafından saklanır.

Sonuçta kontrol Okuyucu Bölümüne geri döner.

13.4.2 Geometrik Bölüm

Bu bölümde üç işleme yordamı vardır:

1. Doğrusal enterpolasyon hareketine bağlı bilgiyi işleyen DLWCIN yordamı
2. Daires enterpolasyon hareketine bağlı bilgiyi işleyen DLWCIR yordamı
3. Tezgah işleme döngüsüne bağlı bilgiyi işleyen DLWCYC yordamı

GODLTA yazılımı ile tanımlanan artışlı hareketten farklı olarak, CLDATA'daki kesici hareketi GOTO/(tek nokta) veya GOTO/(çok nokta) kayıtlarıyla tanımlanır. DLWLIN yordamı genellikle, hareket kaydının DAPP tarafından işlenmesi için kullanılır. Bununla beraber, DLWCIR ve DLWCYC yordamları sırasıyla, ARCSLP/ON ve CYCLE/... yazılımlarının belirlenmesinden, çağrılır. Bağlı değişkenler, bu bölümün işlenmesi süresince ayarlanır ve kontrol işleminden sonra TTM'ye gönderilir.

13.4.2.1 DLWLIN Doğrusal Enterpolasyon Yordamı

Hareket kayıtlarının işlenmesi için kullanılan temel yordamlardan birisi olan DLWLIN, GOTO veya GODLTA kayıtlarından birinin okunması ile çağrılır. Bu yordamın fonksiyonu, doğrusal enterpolasyon hareketinin tanımlanmasında ihtiyaç duyulan parametreleri hesaplamaktır. Bunlar, hedef noktasının mutlak koordinatları, hedef noktasının başlangıç noktasına göre yerleşimi, ilerleme miktarı ve kesicinin hareketi için gerekli zamandır.

Öncelikle, CLDATA'daki parça koordinat sisteminde tanımlanan hedef noktasının mutlak koordinatları, tezgah koordinat sistemine bağlı olarak dönüştürülmelidir. Bu dönüşüm, DLWORG ve DLWROT yordamlarından elde edilen hesaplama sonuçlarına bağlıdır (kesici konum koordinatlarının toplam doğrultmaları gibi). Hedef noktasının dönüştürülen koordinatları DATBUF(1), (2) ve (3)'de

saklanır. Ayrıca, GODLTA yazılımında belirlenen X, Y ve Z yönlerindeki hızlar, sırasıyla DATBUF(7), (8) ve (9)'da ayarlanır. Sonra, eğer programlanılan ilerleme miktarı inç/devir cinsinden belirlenmişse, DLWLIN yordamı ile ilerleme miktarını tekrar hesaplar. Aksi takdirde, belirlenen ilerleme miktarı doğru olarak DATBUF(10)'daki değerle belirlenir. Sonuç olarak, işlem kesicinin başlangıç noktasından hedef noktasına gitmek için gerekli zamanı hesaplar. Bu safhada, kesici hareket hızından farklı bir ilk değeri sahip bir döngü içinde tanımlanırsa, kontrol döngüdeki z konumlarını hesaplamak için DLWCYC yordamına geçer. Sonuç olarak kontrol DAPP'da TIME(1)'de saklanan işleme zamanının hesaplanmasını bitirir ve yordama geri döner.

Özellikle, DLWLIN yordamı, GODLTA ve GOTO kayıtlarını işleyen kontrol DAPP'ın çözümü alanlarından çağrılır. DAPP tarafından bu yordama geçirilen bilgi aşağıdaki kilerden meydana gelir:

1. Başlangıç noktasını tanımlayan DATBUF(4), (5) ve (6)
2. GOTO veya GODLTA yazılımında tanımlanan hedef noktasının x, y ve z koordinatları
3. Parça ve tezgah koordinat sistemleri, kesicinin farkı ve döner tablolar arasındaki dönüşü arasındaki fark için, kesici konum doğrultmaları (Bu değerler DLWORG ve DLWROT yordamları ile hesaplanır).
4. DATBUF(10) VE DATBUF(14)

Bu yordam, ayrıca kesici hareketindeki, SD kontrolcüsünün çözünürlüğündeki farklı eksenlerdeki izin verilen eş zamanlı hareket limitlerini de alır. Bu değerler sırasıyla M01'den M03'e, C04 ve M15'den M18'e kadar verilen sorularla tanımlanır. Bu bilgiler, sınırları aşan tanımlanmış kesici hareketini önlemek üzere DLWLIN tarafından kullanılır. Değerler SD kontrolcüsünün çözünürlüğüne bağlı olarak hesaplanan sonuçları yuvarlatılır. Aynı zamanda ilerleme zamanının hesaplamasında da kullanılır.

Hesaplama sonuçları aşağıdaki değişkenleri tekrar ayarlamak için kullanılır:

1. Tezgah koordinat sistemindeki hedef noktasını tanımlayan DATBUF(4), (5) ve (6)
2. Başlangıç noktasından hedef noktasına olan mesafeyi tanımlayan DATBUF(7), (8) ve (9).
3. Eğer ilerleme miktarı inç/devir cinsinden programlanmış ise, ilerleme miktarı inç/dakika olarak ifade edilmişse, aksi takdirde, ilerleme miktarı girdi olarak verilen değerdir.
4. Kesicinin başlangıç noktasından hedef noktasına gidişi için gerekli ilerleme zamanını tanımlayan TIME(1).

DATBUF içinde kontrol, duruma bağlı olarak çeşitli bölümlere gönderilir. Eğer işlenen kayıt GODLTA veya GOTO/(tek nokta) ise, kontrol işlemin tamamlanmasından sonra TTM'ye gider. Çıktının yazılması veya delikli şeridin hazırlanması için Çıktı Bölümü çağrılır. Sonuçta, yeni bir set kaydın okunması için, kontrol Okuyucu Bölümüne döner. Eğer GOTO/(çok nokta) bir GOTO kaydı işlenirse, kontrol ilk noktanın işlenmesi için gerekli bilgiden sonra TTM'ye, sonra da Çıktı Bölümüne gider. Bilginin çıktısı alındıktan sonra ise, kontrol sonraki noktanın bilgisinin işlenmesi için DLWLIN yordamına geri döner.

Eğer tanımlanan hedef noktası başlangıçla aynı ise, kontrol Okuyucu Bölümüne döner. Bununla beraber, eğer işlenen kayıt birden fazla noktadan ibaretse, kontrol DAPP ana yordamı tarafından tekrar DLWLIN yordamına geri döner.

Eğer işlenen kayıt, işleme döngüsü içinde ise, işleme zamanının hesaplanması durdurulur ve kontrol DLWCYC paket çevrim yordamına geçer. DLWCYC, döngü hareketi için uygun "z" koordinatlarının hazırlanmasından sonra, kontrolü DLWLIN yordamına döndürür.

DLWLIN yordamı ile işleme süresince, hareketin ve ilerleme miktarının tanımlanan sınırlar dışında olup-olmadığının hazırlanması amacıyla birçok test yapılır. Eğer kesici hareketi X, Y ve Z hareketindeki sınırları aşarsa, 6101, 6102 veya 6103'den uygun olan hata mesajları verilir. Programlanmış ilerleme miktarının sıfır veya en büyük değeri aşması durumunda 6116 veya 6107 numaralı hata mesajları oluşturulur. Hataların oluşması durumunda son işleme tamamlanmaz.

13.4.2.2 DLWCIR Dairesel Enterpolasyon Yordamı

Adından da anlaşılacağı gibi, DLWCIR yordamı, dairesele kesici yollarını içeren CLDATA kayıtlarını işler. APT'de tanımlanan bir dairesele hareket, aşağıdaki 3000 ve 5000 kayıt sınıfı setlerine sahiptir:

SURFACE	(Dairesel sürücü yüzey için sembol)
	(Dairenin kabul edilmiş biçimindeki parametreleri)
GOTO/	(Dairesel sürücü yüzey için sembol)
	(İlk noktanın x, y ve z koordinatları)
	(İkinci noktanın x, y ve z koordinatları)

	(Son noktanın x, y ve z koordinatları)

GOTO/(çok nokta) kaydındaki noktalar, SD kontrolcüsünün hesaplama bölümü tarafından hesaplanır. Diğer bir deyişle, APT'deki tanımlanan dairesele yol, belirlenmiş toleransa (Kısım 59'a bakınız) bağlı olarak hesaplanan bir seri doğrusal

enterpolasyon hareket vektörü ile yaklaştırılır. DLWCIR yordamı, NC program çıktısındaki G02 veya G03 dairesel enterpolasyon hareket kodunun oluşumu için ihtiyaç duyulan parametreleri hazırlar. Ayrıca ihtiyaç duyulduğunda kolayca kullanılabilmesi için DATBUF(36)'dan (71)'e kadarki uygun değişkenler saklanır. Eğer dairesel kesici yolunun CLDATA'sı, DLWCIR dairesel enterpolasyon yordamı kullanılmaksızın bir seri doğrusal hareket şeklinde işlenirse, sonuçta parça profili hala tolerans dahilindedir. Bununla beraber, dairesel hareket için NC program parçası çok fazla sayıda G01 kodundan meydana gelir. Fakat bu işlem programı, G02 veya G03 koduyla aynı işleme yolundan tanımlamasını yapan programdan çok uzundur.

NC kodlarında dairesel hareketi tanımlamak için ihtiyaç duyulan parametreler, hedef noktasının koordinatları, kesici yönü (saat ibresi yönü veya tersi) ve yay merkezini tanımlayan gerekli parametre(ler) (yarıçap veya başlangıç noktasından merkeze olan I, J ve K mesafeleri)'dir. Eğer ARCSLP/ON yazılımı tanımlanursa DLWCIR yordamı CLDATA kayıtlarını işlemek için çağrılır. Doğrusal enterpolasyon yordamı, DLWLIN, dairesel hareketin CLDATA kayıtlarının işlenmesi için aşağıdaki sebeplerden dolayı DLWCIR'ın yerine kullanılır:

1. Dairesel hareket yazılımından önce ARCSLP/ON yazılımı kullanılmamıştır.
2. Tanımlanan ilerleme miktarı RAPID'dir.
3. Dairesel hareket, NC kontrolcüsü tarafından izin verilen koordinat düzlemlerinde değildir.
4. GOTO/(çok nokta) kaydındaki nokta sayısı üçten azdır.
5. C03'e verilen cevapla belirlenen en büyük yarıçap, dairesel hareketin yarıçapıyla aşılar veya, bu değer C04 sorusunun cevabıyla belirlenen kontrolcu çözünürlülüğünden küçüktür.

Dairesel hareket içeren CLDATA kayıtları okunduğunda, kontrol uygun ko ayırıcı alana gider. Eğer ARCSLP/ON yazılımı tanımlanırsa, DLWCIR rutin çağrılır. DLWCIR'daki işleme, yukarıdaki beş durumun kontrolü ile başlar. Bunlardan herhangi birinin tetkikinden sonra, kontrol DLWLIN doğrusal enterpolasyon yordamına gönderilir.* Aksi takdirde işlem devam eder. GOTO kaydındaki noktasının koordinatları, DLWORG ve DLWROT yordamlarında hesaplanan sonuçlara göre dönüştürülür ve tezgah koordinat sisteminde tekrar tanımlanır. Daire merkezinin başlangıç noktasından X, Y ve Z farkları da hazırlanır. İşlem bu safhada durmaz, fakat GOTO kaydındaki bir noktayı daha hazırlar. Bu yapıldığı takdirde kontrol ikinci noktasının okunması için Okuyucu Bölümüne geri gönderilir. İşlem

*Yukarıda açıklanan 3 ve 4 durumları için sırasıyla 6119 ve 6122 hata mesajları verilir. Dairesel hareket yarıçapının en büyük değerini aşar veya en küçük değerden küçük ise sırasıyla 6120 ve 6121 hata mesajları verilir.

ilk çeyrek dairenin son noktasına ulaşıncaya kadar devam eder. Sonra X, Y ve Z yönlerindeki farklar ve yerleşim konumları hesaplanıp DATBUF(42)'den (47)'ye kadar olan kısımlarda saklanır. Son noktasının koordinatları, tezgah sınırlamaları ile karşılaştırılır ve kesici konumu X, Y ve Z sınırlarını aşdığı takdirde sırasıyla 6101, 6102 ve 6103 hataları verilir. Aynı şekilde, DLWCIR yordamı sonraki çeyrekler için X, Y ve Z yönlerindeki merkez farklarını ve yerleşimlerini hesaplar, ve DATBUF(48)'den (71)'e kadarki değişkenler ayarlanır. Bu hesaplama, GOTO/(-çok nokta) bilgisindeki son noktasının işlenmesinden sonra oluşturulur. Sonuç olarak, daire parçasının işlenmesi için gerekli olan zaman hesaplanır. İşlemin bitirilmesinden sonra, kontrol MTM'ye gider.

DLWCIR yordamı tarafından ihtiyaç duyulan bilgi aşağıdakilerden ibarettir:

1. DATBUF(4)'den (6)'ya kadarki değişkenler (başlangıç konumunun koordinatları)
2. 3000 CLDATA kayıt sınıfı (daire merkezinin koordinatları ve yarıçapı)
3. 5000 CLDATA kayıt sınıfı: GOTO/(çok nokta)
4. Kesici konumunun X, Y ve Z doğrultmaları, DLWROT ve DLWORG yordamları ile hesaplanır
5. ARCSLP/ON son işlemci komudu
6. LIMIT(1)'den (3)'e kadar, (6) ve (8)'den (11)'e kadar olan değişkenler (tezgah hareketinin sınırları, dairesel hareketin yarıçapı ve ilerleme miktarı)
7. FEDRAT yazılımı ile tanımlanan ilerleme miktarı: Eğer ilerleme değeri inç/dakika cinsinden tanımlanırsa, değişken DATBUF(10)'dur. İlerleme değeri inç/devir olarak tanımlanırsa bir başka değişken kullanılır
8. DATBUF(4) (SPINDL yazılımıyla tanımlanan devir sayısı)

DLWCIR yordamı ile hesaplanan ve ayarlanan değişkenler aşağıdakilerdir:

1. DATBUF(1)'den (3)'e kadar olanlar (dairesel hareketteki hedef noktasının koordinatları)
2. DATBUF(7)'den (9)'a kadar olanlar (X, Y ve Z yönlerindeki toplam yerleşimler)
3. DATBUF(36) (Dairesel enterpolasyon düzlemi)
4. DATBUF(37) (Dairesel hareketin yönü)
5. DATBUF(38)'den (40)'a kadar olanlar (merkez koordinatları)
6. DATBUF(41) (Yarıçap)
7. DATBUF(42)'den (47)'ye kadar olanlar (ilk örneğin ve merkez farkları ve X, Y ve Z yerleşimleri)
8. DATBUF(48)'den (71)'e kadar olanlar (Takip eden çeyrekler için merkez

farkları ve X, Y ve Z yerleşimleri)

9. DATBUF(10) (ilerleme miktarı birimi inç/dakika ise DATBUF(10)'un başlangıçtaki değeriyle aynıdır, aksi takdirde, inç/devir'e ve devir sayısı ile ilgili olarak hesaplanandır)
10. TIME(1) (Kesicinin başlangıçtan hedef konumuna hareketi için gerekli zaman)

Daire merkezinin farklarını oluşturan işlemlerde (APT-AC NC işlemcisi Versiyon 1, Düzey 3) bir hatanın olduğuna dikkat edilmelidir. Bu değerler DATBUF(45)'den (47)'ye, (51)'den (53)'e, (57)'den (59)'a, (63)'den (65)'den ve (69)'den (71)'e kadarki değişkenlerle saklanır.⁹ Problem, bir veya birkaçının negatif olmasına rağmen, hesaplanan değerlerin pozitif olmasıdır. Bu hata, yukarıda bahsedilen DATBUF konumlarının TTM'deki özel bir hesaplama yordamıyla tekrar hesaplanmasıyla doğrultulabilir (Bölüm 14.3.3.2'ye bakınız).

13.4.2.3 CLWCYC Paket Çevrim (Döngü) Yordamı

Birçok deliğin işlenmesini içeren tezgah işlemi durumunda, her delme konumunda kesicinin/işin aşağı-yukarı hareketlerini tanımlamak gerekir. Bu işlem, birçok deliğin söz konusu olduğu durumda oldukça can sıkıcıdır. Bölüm 5'de, çok deliğin işlenmesini tanımlamak için AVOID ikincil kelimesi ile birlikte kullanılan PATTERN yazılımını icra eden bir metod tanımlandı. Bir APT programcısı nokta dizisi ve parametreleri tanımlamak için, NC işlemcisi tarafından oluşturulan kesicinin aşağı ve yukarı hareketlerine ihtiyaç duyar. Birçok durumda delinecek deliklerin derinlik/çap oranı büyüdükçe, matkap talaşı dışarı atmak için belirli aralıklarla dışarı alınmalıdır. Bu hareket tipine KADEMELİ DELME DÖNGÜSÜ adı verilir ve PATTERN yazılımıyla tanımlanamaz. Bu gibi durumlarda tercih edilen, G01 kodlu birçok yazılımı kullanarak delme yerine, G83 delme döngülü bir yazılım oluşturmaktır. CYCLE/... son işlemci komutu, delme başarısız delme, kılavuz çekme ve frezeleme işlemleri için en uygun tanımlamayı yapma imkanı verir.

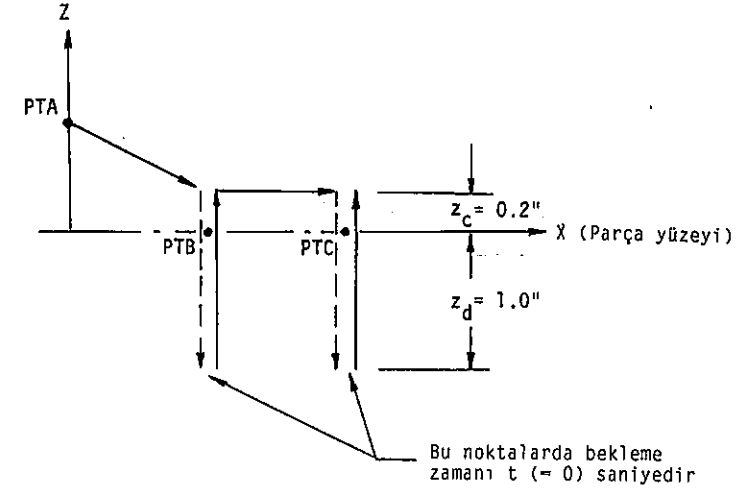
Bir delme veya kılavuz çekme işlemi tanımlayan CYCLE yazılımının formatı aşağıdaki gibidir (Şekil 13-4).

$$\text{CYCLE/} \left\{ \begin{array}{l} \text{DRILL} \\ \text{TAP} \end{array} \right\}, z_d, \left\{ \begin{array}{l} \text{IPM} \\ \text{IPR} \end{array} \right\}, f, z_c, \text{DWL}, t$$

DRILL = Delme döngüsünü tanımlayan son işlemci kelimesi

TAP = Kılavuz çekme döngüsünü tanımlayan son işlemci kelimesi

z_d = Döngüde tanımlanmış (Şekil 13-4) PTB ve PTC noktaları arasındaki delme derinliği



Şekil 13-4 Bir program parçasıyla tanımlanan kesici hareketi.

FROM/PTA
CYCLE/DRILL,1.0,IPM,2.0,0.2
GOTO/PTB
GOTO/PTC
CYCLE/OFF

PTB ve PTC noktaları parçanın üzerindedir.

----- ilerleme değeri 2.0'a programlıyken kesici hareketi

----- yüksek ilerleme değerindeki kesici hareketi

IPM = Yazılımda tanımlanan ilerleme değerinin, f , inç/dakika cinsinden olduğunu ifade eden son işlemci kelimesi

IPR = Yazılımda tanımlanan ilerleme değerinin, f , inç/devir cinsinden olduğunu belirten son işlemci kelimesi

f = İlerleme miktarı

z_c = Tanımlanan PTB ve PTC noktalarına olan açıklık mesafesi

DWL = Kesicinin deliğin tabanında " t " saniye kadar beklemesini sağlayan son işlemci kelimesi

t = Saniye olarak bekleme zamanı.

CYCLE/OFF yazılımı tanımlanan CYCLE fonksiyonunu sona erdirmek için kullanılır. Bunun yanında CYCLE/ON yazılımı kullanılarak sona erdirilmiş işleme döngüsünü tekrar güncelleştirir. Örneğin, Şekil 13-4'de gösterilen döngü için kesici hareketi APT'de aşağıdaki gibi tanımlanabilir.

.....
 (kesici PTA noktasındadır)
 CYCLE/DRILL,1.0,IPM,2.0,0.2,DWL,0.1
 GOTO/PTB
 GOTO/PTC
 CYCLE/OFF

Aşağıdaki program parçası, aynı kesici hareketini tanımlamakta ve CYCLE/ON ve CYCLE/OFF yazılımlarının kullanımını göstermektedir.

.....
 (kesici PTA noktasındadır)
 CYCLE/DRILL,1.0,IPM,2.0,0.2,DWL,0.1
 GOTO/PTB
 CYCLE/OFF
 CYCLE/ON
 GOTO/PTC
 CYCLE/OFF

Bir döngü içinde tanımlanan noktaların aynı seviyede olmadıkları zaman, bir noktadaki geri çekme hareketi, sonraki noktanın "Z" koordinatına bağlıdır. Şekil 13-5'de görülebileceği gibi, eğer $z_{PTC} > z_{PTB}$ ise, PTB noktasındaki geri çekme hareketinin "Z" koordinatı,

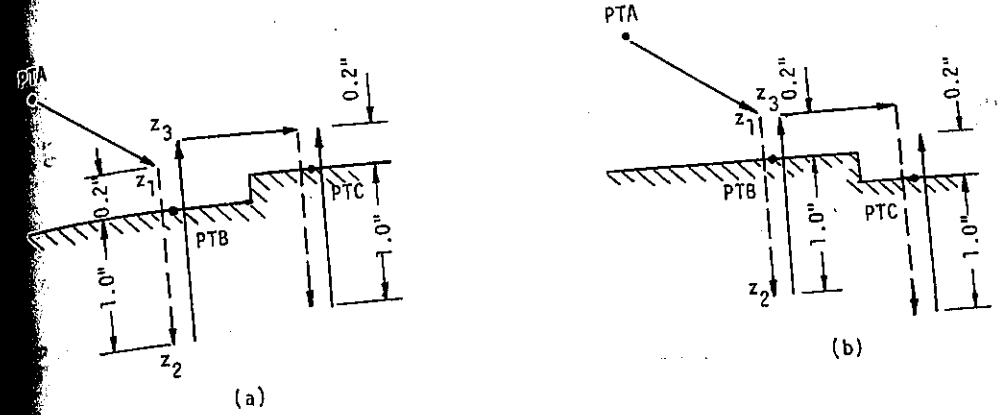
$$z_3 = z_{PTC} + z_c$$

aksi takdirde

$$z_3 = z_1 \text{ 'dir.}$$

Böylece, bu yazılımın fonksiyonunun, döngü içinde tanımlanan noktalardaki ilave Z hareketinin tanımlanması olduğu görülür. Bir noktadaki son konumlar, z_1 , z_2 ve z_3 , derinlik ve başlangıcın mutlak "Z" koordinatları (Şekil 13-5) DLWCYC tarafından hesaplanır ve sırasıyla DATBUF(3), (29) ve (30)'da saklanır.

Bu döngü içinde, GODLTA/z, GOHOME ve GOCLER-CLEARP ile RETRACT-CLSRF yazılım çiftleri de tanımlanabilir. Döngü içinde bu yazılımların kullanım amacı, iki delik arasındaki herhangi engelle çarpmamaları için geri çekme hareketini tanımlamaktır. Farklı geri çekme yazılımına sahip döngülerin hesaplanan sonuçlarının karşılaştırılması Tablo 13-6'da gösterilmiştir. Listedeki sonuçlardan görülebileceği gibi, Z geri çekme hareketi için ilave yazılımlara uygun DATBUF(29) ve (30) konum bilgisi (GOHOME, GOCLER ve RETRACT) bir önceki kayıtlarıyla aynıdır. GODLTA veya GOHOME yazılımları, GOTO'nun



Şekil 13-5 Program parçası ile tanımlanan kesici hareketi
 FROM/PTA
 CYCLE/DRILL,1.0,IPM,2.0,0.2
 GOTO/PTB
 GOTO/PTC
 CYCLE/OFF
 buradaki PTB ve PTC noktaları farklı z koordinatlarına sahiptir.

yaptığı gibi noktadan-noktaya hareketi tanımlar. GODLTA, GOHOME veya GOCLER yazılımın tanımlandığı zaman, DATBUF(3), yazılımla tanımlanan z koordinatıdır. Bununla beraber, RETRACT yazılımına tekabül eden DATBUF(3) değeri sıfırdır. Eğer döngü içinde tanımlanmış ise, CLSRF yazılımının hareket üzerine herhangi bir etkisi yoktur. Bu karşılaştırma, farklı yazılımların işlemesi için DLWCYC'nun hesaplama yordamında, farklı kuralların kullanıldığını gösterir. Bundan dolayı, Takım Tezgaahı Modülü tasarımı süresince bu karşılaştırmanın gözönünde bulundurulması gerekir.

CYCLE yazılımı, aynı zamanda havşalama, gagalama delme, kılavuz çekme ve frezeleme döngülerinin tanımlanması için de kullanılabilir (Tablo 13-7). Kılavuz çekme ve havşalama işlemlerindeki kesici hareketi delme döngüsünün benzeridir. DLWCYC yordamı, havşanın "d" ve "a" parametrelerine bağlı olarak DATBUF(29)'un değerini hesaplar. Gagalama delme döngüsü, zi ve fi parametreleri ile birlikte BRKCHP (talaş kırıcı delme) veya DEEP (derin delik delme) ikincil kelimelerinin kullanımıyla tanımlanır. Frezeleme döngüsü, belirli bir noktada Z eksen boyunca kesicinin aşağı-yukarı hareketini içermez. Bu hareket CYCLE yazılımında olmasına rağmen, programlamadaki kazanç önemli sayılmayacak kadardır.

Bir döngünün ilk kaydı (CYCLE/... yazılımı gibi) okunduğunda, CYCLE yazılımında parametrelerin saklanacağı yardımcı bir dizinin ayarlanması için DLWPP2 yordamı çağırılır. Ayrıca, DATBUF(10) ve (28) ayarlanır. Sonra kontrol

TABLO 13-6 DLWCYC YORDAMI İLE HESAPLANAN KESİCİ KONUM BİLGİSİ (ŞEKİL 13-4'E DE BAKINIZ) PTA: (0.0.1); PTB: (1.0.0); PTC: (2.0.0)

Döngü Hareketini Tanımlayan APT Program Parçası	Tanımlanan Döngü için Kesici Konum Bilgisi					
	DATBUF (1)	DATBUF (2)	DATBUF (3)	DATBUF (29)	DATBUF (30)	
FROM/PTA						
CYCLE/DRILL.1.0.IPM.2.0.0.2	1.0	0	0.2	-1.0	0.2	
GOTO/PTB	2.0	0	0.2	-1.0	0.2	
GOTO/PTC	*	*	*	-1.0	0.2	
CYCLE/OFF						
FROM/PTA						
CYCLE/DRILL.1.0.IPM.2.0.0.2	1.0	0	0.2	-1.0	0.7	
GOTO/PTB	1.5	0	0.7	-0.5	0.7	
GODLTA/0.5.0.0.5	2.0	0	0.2	-1.0	0.2	
GOTO/PTC	*	*	*	-1.0	0.2	
CYCLE/OFF						
FROM/PTA						
CYCLE/DRILL.1.0.IPM.2.0.0.2	1.0	0	0.2	-1.0	1.0	
GOTO/PTB	0	0	1.0	-1.0	1.0	
GOTO/PTC	2.0	0	0.2	-1.0	0.2	
CYCLE/OFF	*	*	*	-1.0	0.2	
FROM/PTA						
CYCLE/DRILL.1.0.IPM.2.0.0.2	1.0	0	0.2	-1.0	0.2	
GOTO/PTB	1.0	0	0.7	-1.0	0.2	
CLEAR/XYPLAN.0.7	2.0	0	0.2	-1.0	0.2	
GOTO/PTC	*	*	*	-1.0	0.2	
CYCLE/OFF						

(Not: Park pozisyonu PTA noktasıdır)

TABLO 13-6 DLWCYC YORDAMI İLE HESAPLANAN KESİCİ KONUM BİLGİSİ (ŞEKİL 13-4'E DE BAKINIZ) PTA: (0.0.1); PTB: (1.0.0); PTC: (2.0.0) (Devamı)

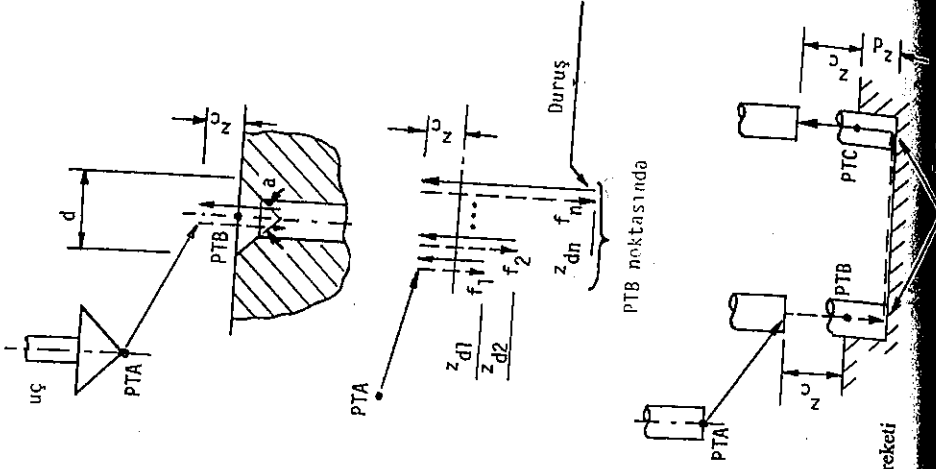
Döngü Hareketini Tanımlayan APT Program Parçası	Tanımlanan Döngü için Kesici Konum Bilgisi					
	DATBUF (1)	DATBUF (2)	DATBUF (3)	DATBUF (29)	DATBUF (30)	
FROM/PTA						
CYCLE/DRILL.1.0.IPM.2.0.0.2	1.0	0	0.2	-1.0	0.2	
GOTO/PTB	1.0	0	0	-1.0	0.2	
CLRSRF/0.7	2.0	0	0.2	-1.0	0.2	
RETRACT	*	*	*	-1.0	0.2	
GOTO/PTC						
CYCLE/OFF						
FROM/PTA						
CYCLE/DRILL.1.0.IPM.2.0.0.2	1.0	0	0.2	-1.0	0.7	
GOTO/PTB	1.0	0	0	-1.0	0.7	
RETRACT	2.0	0	0.2	-1.0	0.2	
GOTO/PTC	*	*	*	-1.0	0.2	
CYCLE/OFF						

* ANULL

TABLO 13-7 HAVŞALAMA, ÇOK PASODA DELME VE FREZELEME İŞLEMLERİNİ TANIMLAYAN DÖNGÜ YAZILIMININ FORMATI

1. Havşalama döngüsü: Yazılım formatı:	CYCLE/CSINK, $d, z, \{IPM\}, f, dc, DWL, t$
Şekilde görülen kesicinin hareketi aşağıdaki program parçasıyla tanımlanır:	
FROM/PTA	
CYCLE/CSINK, 0.5, 90, IPM, 1.0, 0.6, DWL, 1.0	
GOTO/PTB	
CYCLE/OFF	
2. Çok pasolu-delme (DEEP veya BRKCHP) döngüsü: Yazılım formatı:	CYCLE/ (DEEP/BRKCHP), $z, a_1, \{IPM\}, f_1, z, a_2, \{IPM\}, f_2, \dots, z, a_n, \{IPM\}, f_n, z, a_c, DWL, t$
Şekilde görülen kesicinin hareketi aşağıdaki program parçasıyla tanımlanır:	
FROM/PTA	
CYCLE/DEEP, $z, a_1, IPM, f_1, z, a_2, IPM, f_2, \dots, z, a_n, IPM, f_n, z, a_c, DWL, t$	
GOTO/PTB	
CYCLE/OFF	
3. Frezeleme döngüsü: Yazılım formatı:	CYCLE/MILL, $z, a, IPM, f, IPM, f_1, z, a, DWL, t$
Şekilde gösterilen hareket aşağıdaki program parçasıyla tanımlanır:	
FROM/PTA	
CYCLE/MILL, 0.2, IPM, 1.0, IPM, 3.0, 0.5, DWL, 1.0	
GOTO/PTB	
GOTO/PTC	
CYCLE/OFF	

Not: - - - - - → Programlanan ilerleme miktarındaki kesici hareketi → Hızlı ilerleme miktarındaki kesici hareketi



Okuyucu Bölümüne döner. Döngü içindeki GOTO bilgisi okunur. İlerleme hareketinin RAPID'den farklı tanımlandığı hareketle, DLWCYC yordamının çağrılmasında, kontrol DLWLIN yordamına gider. DLWCYC paket döngü yordamı her bir noktanın "Z" değerlerini DATBUF(3), (29) ve (30) değişkenleri şeklinde ayarlar. İlerleme değeri ve bekleme zamanı sırasıyla DATBUF(10) ve (19), döngü durumu DATBUF(31) (veya LOGCAL dizisi) şeklinde ayarlanır. Bu işlemlerden sonra kontrol, DAPP ana yordama geri gönderilmek üzere DLWLIN'e döner. Bilginin daha ileri safhada işlenmesi için, APT programındaki MACHIN yazılımı ile tanımlanan TTM çağrılır. Bilginin çıktısının alınmasından sonra, kontrol Okuyucu Bölümüne geri gider. Eğer döngü, gagalama delme ise (DEEP veya BRKCHP kelimeleri ile tanımlanan) başka Z değerlerini ayarlaması için DLWCYC yordamı tekrar çağrılır, ve işlem yukarıdaki gibi tekrar ettirilir. Aksi takdirde, yeni bir CLDATA kaydı okunur.

Özet olarak DLWCYC'ye aktarılan bilgiler aşağıdaki gibidir:

1. DLWLIN tarafından hesaplanan DATBUF(3).
2. CYCLE yazılımında tanımlanan boşluk ve derinlik değerleri.
3. DLWROT ve DLWORG yordamları tarafından hesaplanan, X, Y ve Z yönlerindeki kesici konumlarının doğrultmaları.
4. CLDATA 5000 kayıt sınıfı: GOTO/(nokta)
5. Bir önceki FEDRAT veya CYCLE yazılımında tanımlanan ilerleme miktarı.
6. RAPID ilerleme miktarı: DATBUF(11)
7. Programlanan devir sayısı: DATBUF(14)
8. CYCLE yazılımında tanımlanan bekleme zamanı.

DLWCYC yordamı tarafından hesaplanan ve ayarlanan değişkenler aşağıda verilmiştir:

1. DATBUF(3) (DATBUF(3) ve boşluk değerinin girdi miktarı)
2. DATBUF(29) (Delme derinliği)
3. DATBUF(30) (Geri çekme hareket yüksekliği)
4. Döngü durumunu ifade eden LOGCAL(1)-(4) değişkenlerini içeren DATBUF(31).
5. DATBUF(10) (Talaş kaldırma ilerlemesi)
6. DATBUF(19) (Bekleme zamanı)
7. TIME(1) (Döngü hareketi için işleme zamanı)
8. Eğer tezgah kesicinin z koordinatlarını kontrol için tamburlar kullanıyorsa, DATBUF(32) ve (33).

DATBUF(3), (29) veya (30) "z" koordinatlarından herhangi biri Z yönünde hareket sınırlarını aşarsa, 6106 hata mesajı verilir. Boşluk sayısı veya döngü hareketinin kontrolü için gerekli tambur derinlikleri tezgahdaki tambur sayısından (M28 ve M29 sorularına verilen cevaba bakınız) fazla olursa; 6108 ve 6109 hata komutları belirir.

13.5 ÇIKTI BÖLÜMÜ

Çıktı Bölümü, DAPP içindeki son işlemci adımını icra eder. Bir set CLDATA genel ve tezgah-bağımlı işleme gittiğinde, Çıktı Bölümü, MTM tarafından çağrılır. Çıktı Bölümünün fonksiyonları şöyle sıralanabilir: İşlenen bilgiyi ve mesajı, belirli bir şekilde yazdırmak. Çıktı programını, delikli kartlara, manyetik teyp veya diskere, veya delikli şeritlere (ISO veya EIA kodunda) kaydetmek. Kayar nokta sayıları ve karakterleri; EBCDIC* karakterleri, delikli kartlar ve ISO veya EIA delikli şerit şeklindeki formatlara dönüştürmek. Delikli çıktı için istenen format, MACHIN yazılımındaki ilk parametre ile seçilir (Bölüm 6.12.2'ye bakınız).

Çıktı Bölümü, bir esas ve birkaç alt yordamdan ibarettir. Esas yordam, girilen bilgisini istenen çıktı formatına dönüştürecek alt rutini seçen Çıktı Bölümündeki kontrolü yönetir. Bu bölümde bulunan diğer bir yordam ise DLWDER'dir. Bu son işlemciye hata mesajlarını çıktı olarak verir.

Çıktı Bölümünde çağrılan üç farklı durum vardır ve kontrol bunlara bağlı olarak hareket eder. CLDATA, genel ve tezgah-bağımlı işlemeyen başarılı bir şekilde geçmişse, MTM, Çıktı Bölümünü çağırır. MTM tarafından belirlenmiş isteklere göre çıktı alındıktan sonra (MTM'deki OUTCTL değişkeniyle), Çıktı Bölümü kontrolü TTM'ye geri gönderir.

Eğer okunan CLDATA, son işlemci komutu INSERT veya PPRINT ise, Çıktı Bölümü, DLWPP2 yordamı tarafından çağrılır. Açıklama yazısı çıktı listesine yazılır veya INSERT yazılımının kullanımı ile delikli kart veya şeride delinir. Kontrol DLWPP2 yordamına döner ve yeni bir set CLDATA bilgisinin okunması için Okuyucu Bölüm çağrılır.

Aynı zamanda, Çıktı Bölümü, Genel İşleme Bölümü ve MTM tarafından gerçekleştirilen işleme süresince oluşan hatalar durumunda da çağrılır. Hata mesajları DLWDER yordamı tarafından yazdırılır ve kontrol çağırma yordamına geri döner.

Çıktı Bölümünden genellikle iki çeşit çıktı alınır: Yazılı liste ve kart veya şerit şeklindeki delikli kart. Yazılı listede, işlenen NC kodları, gerekli işleme zamanı, NC şeridin uzunluğu, ve eğer varsa hata mesajları bulunur. DATBUF'daki seçilen değişkenlerin değerleri de işlem sonucunun kontrol edilebilmesi amacıyla çıktıda bulunabilir. DAPP'ın yazılı çıktısının formatı, Bölüm 15'de açıklanmıştır.

* Extended Binary Coded Decimal Interchange Code'nun kısaltılmışı

Delikli çıktı, aşağıdaki sıralı öğelerle başlar:

1. Eğer P02 sorusuna verilen cevap YES ise, PARTNO yazılımında tanımlanan kişinin okuyabileceği açıklama yapısı.
2. PLABEL yazılımıyla tanımlanan, delikli şerit, parça veya tasarım değişikliği seri numarasına bağlı kişinin okuyabileceği yazı.
3. P01 sorusuna verilen cevapla tanımlanmış ise, şerit sürücüsü.
4. C06 sorusuna verilen cevap YES ise, şeridin geri döndürmeyi durdurma kodu.
5. C07 sorusuna verilen cevap YES ise, blok sonu kodu.

Bu unsurları, son işlemci tarafından oluşturulan NC kodundaki parça programı izler. Eğer DAPP tarafından herhangi bir hata tesbit edilirse veya istenmezse, delikli çıktısı alınmaz. Bölüm 6'da belirtildiği gibi, hatanın da çıktısının alınması için DAPP, RESET yazılımı ile yönlendirilir.

PROBLEMLER

- 13.1 İlk Değer Atama Bölümü ile Düzenleme Bölümü fonksiyonları arasındaki farkı açıklayınız.
- 13.2 Tablo 13-1'de listelenmiş DATBUF dizisinin elemanlarını güncelleştirmenin sebeplerini açıklayınız.
- 13.3 DAPBUF dizisindeki bazı elemanlar, DAPP'ın Düzenleme Bölümü tarafından güncelleştirilmemiştir. Özel bir son işlemcinin işlem gereklerine göre tamamlanmış CLDATA kayıtlarının bir takımının okunması işleminden sonra güncelleştirilmiş ise bu elemanların revizyonlarını nasıl gerçekleştiririz.
- 13.4 Parametresiz son işlemci kelimeleri (komutları) rutin olarak DLWPP1 tarafından işlenir. İşleminin tamamlanmasından sonra kontrol DAPP'ın üç ayrı bölümüne gönderilir. Her mümkün durum için bir örnek veriniz, kontrol akışını belirleyiniz ve örneğimizde, kontrolün belirtilen bölüme gönderme sebebini açıklayınız.
- 13.5 MACHIN deyimindeki üçüncü ve dördüncü parametreler, APT programından son işlemciye faydalı bilgiyi iletmek için kullanılabilir. Bu parametrelerin mümkün kullanımlarını listeleyiniz.
- 13.6 CYCLE son işlemci komutunu kullanarak problem 8.9'da belirtilen delme hareketini tanımlayınız.
- 13.7 Tablo 13-4 ve 13-5'de listelenmiş her son işlemci kelimesi (komutu) için son işlemci tarafından işlenmesi gereken NC kodlarını veriniz.
- 13.8 Makinanın birinde bir hata bulunduğu bağımsız işlem yordamlarında ve karşılık gelen çıktı (bu DATBUF elemanıdır) yanlış-doğru NC kodlarının son işlemci tarafından hâlâ üretilebildiği DATBUF elemanlarını düzeltme metodunu öneriniz.

Bölüm 14

Tezgağı-Bağımlı İşleme Yordamı: Takım Tezgağı Modülü

DAPP'a bağılı son işlemcinin takım tezgağı modülü (MTM-Machine Tool Module, MTM), Genel İşleme Bölümü tarafından hazırlanan ve daha sonra Çıktı Bölümüne aktarılan ve NC kodları olarak çıktısı verilen daha ileri veri işlemeyi icra eden bir alt programdır.

CLDATA grubu genel işlemenden geçtikten sonra, DATBUF dizisinde saklanır. MTM çağrıldığı zaman, önce her veri elemanı NC tezgağının karşılık gelen sınırları içinde olduğundan emin olmak için kontrol edilmelidir. Eğer veri elemanı doğrudan doğruya NC kodlarının çıktısı için kullanılmazsa, tekrar hesaplanmalı veya işlenmelidir. Genellikle Okuyucu Bölümü tarafından okunan CLDATA grubu birden fazla tek veri elemanı içerir; bunun bir veya daha fazla NC blokları olarak çıktısı verilebilir. Böylece MTM, bir NC bloğunun (deyiminin) içeriğini de tespit etmelidir. Verinin çıktı kısmına aktarılmasından önce, NC kodun doğru formatı da MTM tarafından ayarlanmalıdır. Ayrıca MTM, APT programında tanımlanmayan, fakat NC'li tezgağı tarafından gerekli olan özel deyimlerin türetilmesinden de sorumludur. Onun için MTM'nin fonksiyonları aşağıdaki gibidir:

1. NC'li tezgağın izin verilen sınırları içinde olduğundan emin olmak için kesici hareket ve tezgağı işlemi ile ilgili verileri gözden geçirmek
2. Verileri NC'li tezgağın denetleyici sistemine kabul edilebilir olan ilgili kodlara çevirmek ve dönüştürmek
3. Bir NC bloğun içeriğini tespit etmek
4. NC'li tezgağın ihtiyaçlarına izin veren kızakların emniyetli hızlanması veya yavaşlaması ve bir köşeden öteye gitme gibi hareket komutlarını geliştirmek
5. NC'li tezgağı için gerekli olan özel deyimleri türetmek

DAPP Genel İşleme Bölümü tarafından hareket veya takım tezgağı işlemi tanımlayan bir grup CLDATA işlendikten sonra MTM çağrılır. İşlemeden sonra, CLDATA verisinin okunan bir seti, bir NC bloğunda çıktı olarak bulunmayan kodlardan (örneğin iki M kodu) meydana geldiğinde, CLDATA'nın bir kısmı çıktı olarak alınır. Sonra MTM geri kalan bilginin işlenmesi için tekrar çağrılır.

Bu bölümde, önce bir MTM'nin genel yapısı sunulacaktır. Sonra istenen fonksiyonları gerçekleştirmek için kullanılan yöntem açıklanacak ve MTM'nin farklı kısımlarının yapısı anlatılacaktır. Son olarak, MTM'de işlem akışı ve DAPP'a dayalı son işlemci açıklanacaktır.

14.1 BİR TAKIM TEZGAĞI MODÜLÜNÜN GENEL YAPISI

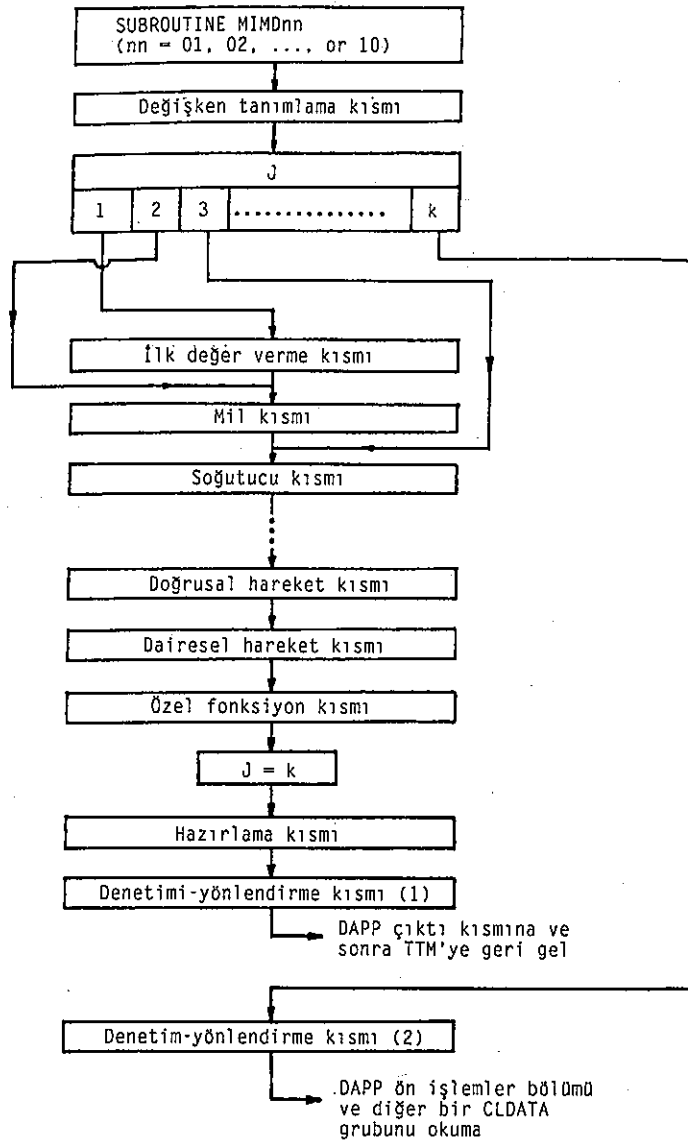
DAPP'a dayalı bir son işlemci için MTM, son işlemci tasarımcısı tarafından yazılır ve işleme için uygun olduğunu düşündüğü bir yolla tasarlanabilir. Fakat, kendilerinin vasıtası ile MTM'nin DAPP ile iletişim kurduğu değişkenlerin isimleri ve formatları takip edilmelidir. Genellikle MTM'nin tasarımı esnasında ele alınması gereken değişkenler Kısım 12.1'de açıklanan değişkenlerden ibarettir. Genel olarak, bir MTM beş kısma bölünebilir (Şekil 14-1):

1. Kendilerinin vasıtası ile MTM'nin DAPP ile iletişim kurduğu COMMON değişkenlerini tanımlamak için ve MTM'ye özel değişkenleri tanımlamak için değişken tanımlama kısmı,
2. MTM'ye özel değişkenlerin, dizilerin ve parametrelerin her birisinin kendi ilk değerlerini ayarlamak için ilk değer verme kısmı (Bu kısmın diğer bir fonksiyonu bir NC programının başında özel bazı NC deyimlerinin çıktısını vermektir.)
3. NC kodlarının formatını ve bir NC bloğun içeriğini tespit etmek için işleme kısmı (Bir çeşit CLDATA olan her bir işleme birkaç alt kısımdan oluşur.)
4. Bir NC bloğun içeriğini son duruma getirmek ve DAPP'ın Çıktı Bölümüne devredilecek olan NC kayıtlarını tutmak için hazırlama kısmı.
5. MTM'yi terkettikten sonra denetimi yönlendirmek için denetimi yönlendirme kısmı.

MTM, DAPP programının bir alt programı olduğu için onun ilk deyimini alt programın ismini tanımlayan FORTRAN deyimini olmalıdır. DAPP'ta bir MTM için kullanılabilen isim MTMD_{nn}; burada nn, 01,02,..., veya 10 olabilir. Böylece bir MTM'nin ilk deyimini şu olmalıdır:

```
SUBROUTINE TTMDnn
```

buradaki *nn*, E02 sorusuna (Bölüm 12.2.3'e bakınız) olan cevap ile aynı olmalıdır.



Şekil 14-1 MTM'nin genel yapısı.

İlk deyimden sonra, yukarıda açıklanan beş kısım, peşpeşe listelenmiştir.

Değişken tanımlama kısmı iki parçaya bölünebilir. Birinci parça, MTM ve DAPP arasında iletişim bağlantısı olarak görev yapan değişkenleri tanımlayan deyimleri içerir (Şekil 14-2). Bunlar bir DAPCOM etiketli COMMON bölgesi, ilgili BLOCK DATA programı (Kısım 12.3 ve Şekil 12-4'e bakınız) ile iletişim

```

IMPLICIT
COMMON /DAPCOM/
REAL*8 (A-H,O-Z)
A ANULL ,CLEARS (3) ,CSURF ,CORREC (3) ,FEDIPR
B ,FEDMAX ,INBUF (252) ,ORGX ,ORGY ,ORGZ
C ,PNO (6) ,PTNOIP (11) ,RADI ,RDRUM (20) ,TABPOS
D ,THETA ,TINATH (4) ,WORKR8 ,ZDRUM (20)
E ,AUXARR (6) ,CLKARG ,CLKHRS ,CLKMIN ,CLKSEC
F ,DHFED (10) ,DPHOL (10)
G ,CLKSW6 ,CONTRL ,DUMREC ,ERRCOD ,IDIREC
H ,LENSW ,IFED1 ,IFED2 ,IFIPR ,IJSEQ
I ,ILIM ,IERCNT ,IERINC ,IERROR (20) ,INCLK
J ,INSUN ,IPLAB ,IPOINT ,IRECNO ,ISKIP
K ,ITAPL ,IWORDS ,JASGN ,JEND ,JLINE
L ,JP ,LNCLRC ,NBSTEP ,NOSETP ,NOWRD
M ,NSST ,NSUBCL ,OUTCTL ,OUTRQD ,PASS7
N ,PGCNT ,PKTARG ,PRTLIN (120) ,PUNBUF (144) ,PUNTAP
O ,RESET5 ,SEQN7 ,SEQINC ,ZPCTL1
REAL*8
REAL*4
A , ,CLKSEC
B , ,CONTRL ,DUMREC ,ERRCOD
C ,OUTCTL ,OUTRQD ,PASS7 ,PGCNT ,PRTARG
D ,PUNTAP ,RESET5 ,SEQN7 ,SEQINC ,ZPCTL1
E ,PRTLIN ,PUNBUF
F ,ORG (3)
G ,ORGX
H ,DP
I ,CL
J ,DW
REAL*8
EQUIVALENCE
REAL*4
EQUIVALENCE
LOGICAL*1
EQUIVALENCE
COMMON /QUEST/
A ,CORTRS (3) ,DATBUF (85) ,HOME (3) ,LIMITS (11) ,MAXRAD
B ,PPCODE ,PPPARM ,TPLIMT ,ULINE0 (15) ,ULINE1 (15)
C ,ULINE2 (15) ,XCONVT ,YCONVT ,ZCONVT ,XTRA (2)
D ,TIME (8) ,ULINE3 (15) ,ULINE4 (15)
E ,CLDRMS ,CNTOUR ,DISKID ,DPDRMS ,DRUMZE
F ,IGOHOM ,IGOTAB ,IRPMOD ,IZHOR ,LDRLN
G ,LENSW ,NLINE ,OUTGRD (20,10) ,R10 ,STRISQ
H ,SIMOSW (4) ,TABLGO ,ZMOD (2) ,Z10
I ,CLKPRT ,EOB ,IEOB ,IPART ,IRSTO
J ,LCHAR (2) ,TABCOD ,TAPPRT ,TARMET
REAL*8
REAL*4
A , ,LIMITS ,MAXRAD
B , ,TIME ,ULINE3 ,ULINE4
C ,CLDRMS ,CNTOUR ,DISKID ,DPDRMS
D ,DRUMZE ,OUTGRD ,R10 ,STRISQ
E ,TABLGO ,ZMOD ,Z10
F ,CLKPRT ,EOB ,IEOB ,IPART
G ,IRSTO ,LCHAR ,TABCOD ,TAPPRT ,TARMET
H ,ZDEPTH ,DATBUF (29) ,ZFINAL ,DATBUF (30)
I ,CONVEN (3)
J ,CONVEN (1) ,XCDIG ,XCONVT ,YCDIG ,YCONVT ,ZCDIG ,ZCONVT
A ,ASSIGN (2)
B ,ASSIGN (1) ,DATBUF (20) ,CYCTYP (2)
C ,LOGCAL (6) ,OUTSUX (40) ,OUTSUX (1) ,DATBUF (28)
D ,LOGCAL (1) ,DATBUF (31) ,OUTSUX (1) ,OUTGRD (1,1)

```

Şekil 14-2 TTM'nin değişken tanımlama kısmında tanımlanan deyimler. Bu deyimlerin tanımladığı değişkenler aracılığı ile MTM DAPP ile iletişim kurar.

için kullanan QUEST etiketli deyim ve değişkenleri tanımlayan diğer bazı deyimler ve hem DAPP hem de MTM tarafından kullanılan ortak saklama yerlerinden ibarettir. Normal olarak, bu deyimler değiştirilmemelidir. Onun için, Şekil 14-2'de gösterildiği gibi bu kısmın ilk parçası her MTM için aynı olmalıdır.

Değişken tanımlama kısmının ikinci parçası, sadece bu özel MTM içinde kullanılan değişkenleri tanımlayan deyimlerden ibarettir. Genellikle bir MTM için gerekli değişkenlerin bir tipi o andaki NC kodlarının kaydını tutan değişkenlerdir. Bir sonraki NC bloğu işlem görürken bu değişkenler işleme, hesaplama veya her ikisi için referanslar olarak kullanılır.

Bu kısımda, MTM'nin her çağrılışında bu değişkenlere aynı ilk değerler tekrar

ilk değer olarak verilmelidir. Bu kısmın bütün bir örneği 15. Bölümde verilmiştir. Değişken tanımlama kısmından sonra, denetim akışını yönlendiren bir deyim olmalıdır. Normal olarak bu amaç için kullanılan FORTRAN deyimi:

GOTO (n1,n2,...,nk),J

burada n1,n2,...,nk deyim etiketlerini temsil eden işaretli tam sayı sabitleridir. J, indisi olmayan tam sayı değişkenidir.

MTM'nin ilk değer verme kısmı iki fonksiyona sahiptir. Birincisi, tüm APT programının işlenmesi esnasında sadece bir kere ilk değerine getirilmesi gereken değişkenler ve diziler için ilk değerleri ayarlamaktır. İkincisi bir NC programın başlangıcında NC'li tezgah tarafından gerekli özel deyimleri üretmektir (örneğin 2. Bölümde açıklandığı gibi tornada kesiciyi referans noktasına geri döndürmek kullanılan deyim). Onun için bu kısım, bir APT programı işlenmesinin başlangıcında sadece bir kere girilmelidir. Bu kısım sonunda, işlem görmüş GOTO deyimindeki J parametresini 1'den (1 değeri ilk değer verme kısmına uyarsa) başka bir değere ayarlayarak gerçekleştirilebilir (Şekil 14-1'e bakınız).

İşleme kısmı, okuma verisinin incelenmesi ve çevrilmesi, tezgah dinamiğiyle uyum sağlaması için belirli hareket komutlarının geliştirilmesi ve bir NC bloğunun içeriğinin tespitini içine alan bir MTM'nin ana fonksiyonlarını icra eder. Bu fonksiyonların gerçekleştirilmesi için yöntem Bölüm 14.3'te açıklanmaktadır.

Bir CLDATA grubu, DAPP'ın Genel İşleme Bölümü tarafından MTM'ye aktarıldığında, bu DATBUF dizisinde dönüşüme tabi tutulur ve saklanır. MTM'nin işleme kısmı her biri sadece bir çeşit veri işlemeye muktedir bir kaç alt kısımdan oluşur (iş mili kısmından Şekil 14-1'deki özel fonksiyon kısmına kadar). İşleme esnasında, bir grup CLDATA (veya DATBUF değişkenleri) bir kısımdan diğerine işleme kısmı aracılığı ile geçer. Her alt kısım, sadece kullanmak için tasarlanan DATBUF konumlarını işler. Bu kısım geçtikten sonra, bütün DATBUF elemanları uygun NC kodlarına çevrilir. Bununla beraber, bir NC blokta izin verilmeyen bir kodun bir işlem alt kısmında bulunduğu tespit edilirse, bütün veriler işlenmeden önce, denetim bu kısmı terk edebilir (bu durum Şekil 14-1'de gösterilmemektedir). Böylesi bir durumda, denetim, hazırlama kısmı verilerin çıktısına aktarılır. Sonra denetim MTM'ye tekrar girer; burada geri kalan verilerin işlenmesine kaldığı yerden devam eder. Bütün alt kısımlar icra edildikten ve bütün okuma veri grubu işlendikten sonra MTM, J değişkenini k'ya ayarlar (Şekil 14-1'e bakınız): MTM'ye tekrar girdiğinde bu, denetimi denetim yönlendirme kısmının ikinci kısmına yönlendirir. Sonuç olarak, denetim DAPP ön işlemler kısmına gider ve DAPP diğer bir CLDATA grubunu okur.

MTM'yi terkettikten sonra denetim akışının yönü, OUTCTL değişkeni tarafından kabul edilen bir değerle belirlenir. Denetim yönlendirme kısmı OUTCTL

nin uygun değerini ayarlar böylece Denetim doğru olarak yönlendirilebilir. Bu bölümün diğer bir fonksiyonu, MTM tekrar çağrıldığında yeniden ayarlanmayan veya ilk değeri tekrar verilmeyen değişkenleri tekrar ayarlamaktır.

Denetim dallandırma deyimindeki J değişkeninin MTM içinde denetim akışının yönlendirilmesi için kullanıldığı yukarıda yapılan açıklamadan anlaşılmalıdır. Diğer taraftan OUTCTL değişkeni denetimi DAPP içinde yönlendirir. Onun için bu iki değişkenin mantıksal birleşimi MTM ve DAPP içinde istenilen işlem akışını tespit eder.

14.2 MTM'NİN İLK DEĞER VERME KISMI

Önceden açıklandığı gibi, MTM'nin ilk değer atama bölümünün fonksiyonlarından birisi, sadece bir kere ilk değere getirilmesi gereken (ilk defa MTM çağrıldığında) değişkenlerin ilk değerlerini ayarlamaktır. Örneğin, değişkenler, çıktı listeleme dosyasında yazılan DATBUF konumlarının tanımlamasını içerebilir (N1, ISTART ve LENGTH değişkenleri için Kısım 13.1'e bakınız). O anda işlenen sonuçları (kodları) kaydetmek için kullanılan değişkenler veya dizi, bu kısımda ANULL (girdi yok) olarak ilk değeri verilmelidir. MTM'nin tasarımına bağlı olarak, bu kısımda ilk değeri verilmesi gereken diğer değişkenler de olabilir.

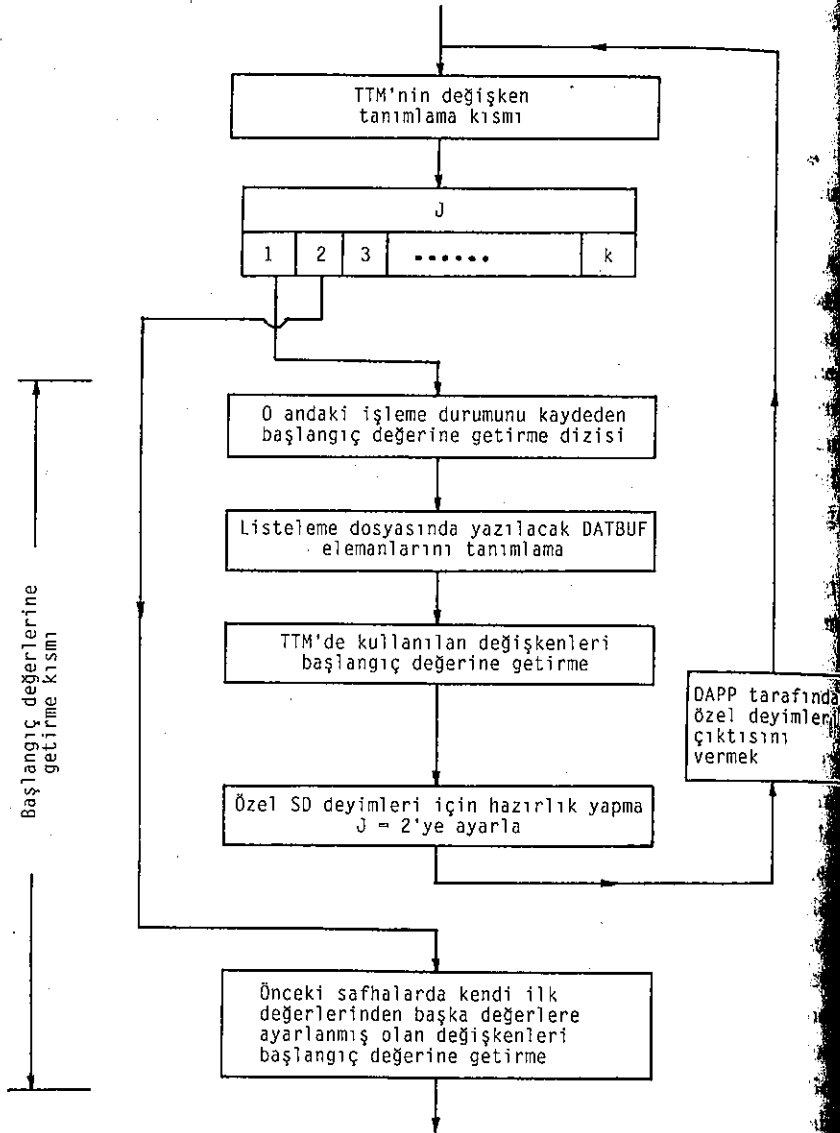
Eğer uygunsuz, bir son işlemci tasarımcısı, MTM'nin değişken tanımlama kısmında bu değişkenlere ilk değerini vermek için icra edilemeyen DATA deyimini kullanmayı seçebilir.

Bu bölümün diğer fonksiyonu, NC'li tezgah tarafından gereken özel deyimleri ayarlamak ve çıktı olarak vermektir. Örneğin, aşağıdaki NC blok, Cadillac NC-100 torna/FANUC 6T denetleyici sistem için bir NC programda ilk deyim olarak gerekmektedir:

G28U0W0

Bu NC bloğu üretecek FORTRAN program parçası aşağıdaki gibi olabilir:

```
DATBUF(73)=28
DATBUF(74)=0
DATBUF(75)=0
OUTGRD(2,10)=73
OUTGRD(11,10)=74
OUTGRD(12,10)=75
OUTCTL=6
J=2
RETURN
```



Şekil 14-3 MTM ilk değer atama bölümünün genel yapısı

İlk üç deyim kodları çıktı olarak saklamak için kullanılır. 4.-6. deyimler 73, 74, 75 DATBUF konumlarına işaret eden G, U ve W kodları ile ilgilidir. Son OUTCTL kontrol yönlendirme değişkeni 6'ya ayarlanır; bu OUTGRD tarafından tanımlanan formata bağlı olarak NC kodlarının çıktı olarak verilmesi gerektiğini gösterir. MTM'de kontrol akışını yönlendiren GOTO deyiminde

değişkeni 2'ye ayarlanır; bu MTM tekrar çağrıldığında kontrolün bir sonraki kısma gitmesi gerektiğini gösterir. RETURN deyimi, denetimi tanımlanan NC bloğu çıktı olarak vermek için denetimi DAPP'a geri gönderir. Sonra denetim MTM'ye tekrar girer ve bu program parçasından sonra değişken tanımlama bölümünden diğerine kısma doğrudan gider. Sonra bu program parçasında kullanılan değişkenlere tekrar ilk değer atamaları yapılır.

Bütün ilk değer atama bölümünün yapısı Şekil 14-3'te gösterilmektedir. Bu bölüm icrasını tamamladıktan sonra, tezgahla ilgili değişkenler ilk durumlarına ayarlanır ve NC'li tezgah için gerekli NC deyimleri çıktı olarak verilir. Böylece MTM, okunan CLDATA'yı işlemeye hazır hale gelir.

14.3 MTM'İN İŞLEME KISMI

Okunan CLDATA çeşitli verilerden ibarettir. Genellikle, iki tipe bölünebilir: Tezgah işlemi ile ilgili ve kesme hareketi ile ilgili veri. Bu bölümün girişinde de açıklandığı gibi, bu veriler bir veya birkaç işleme tipine tabi tutulmalıdırlar. Çünkü her türlü veri, farklı çalışma işlemlerini gerektirir. MTM'nin işleme bölümü daha sonra, fener mili dönme hızı ve yönü, soğutucu, kesici, doğrusal ve dairesel enterpolasyon hareketi ve özel tezgah operasyon fonksiyonları gibi çeşitli verileri işleyen alt kısımlara bölünmelidir.

14.3.1 Bir NC Bloğun İçeriğini Tespit Eden Yöntem

İşleme bölümü tasarlanmadan önce, bir NC blokta kurulabilen kodların çeşitlerini tespit eden kurallar belirlenebilmelidir. Örneğin, FANUC 6T ve 6MB denetleyiciler için, farklı cins kodlar, bir NC blokta rastgele bir sırada belirlenebilir. Fakat, 2. Bölümde açıklanan G kodlarının bazıları hariç aynı cinsten olan kodların, bir blokta olmalarına izin verilmez. Bu durumda iki kural vardır:

Kural 1: Bir NC blokta izin verilen G kodları hariç, aynı cins iki kod bir NC blokta tanımlanamaz.

Kural 2: Bir NC bloktaki kodlar farklı cinsten olmalıdır.

Birinci kural, FANUC sistemi için NC programlamada ne yapılacağına izin verildiğini belirtir. Bunu gerçekleştirmek için iki veya daha fazla OUTGRD grup elemanları G kodunu tanımlamak için kullanılmalıdır. Örneğin, eğer MTM iki G kodunun bir SD bloğunda çıktısını alabilecek şekilde tasarlanırsa, OUTGRD(1,i) ve OUTGRD(2,i), $i = 1, 2, \dots, 9$ Tablo 12-2'deki dipnotta gösterilen G kodunu tanımlamak için kullanılabilir. İşleme programında, şunlar yazılabilir:

G1=OUTGRD(1,10)

ve

G2=OUTGRD(2,10)

G kodu olarak çıktı olabilen bir veri elemanı bir DATBUF konumunda işlenip saklandığı zaman DATBUF(73) adını alır, sonra G1 = 73 ayar yapılır. Böylece DATBUF(73) elemanındaki verinin bir G kodu olarak çıktısı alınabilir. Eğer sonraki veri elemanı DATBUF(74)'te saklanırsa ve diğer bir G kodu olarak çıktı alınırsa önce bu G kodunun bir NC blokta önceki ile birlikte çıktısının alınmaması için G2 = 74 ayar yapılabilir. Bir grup OUTGRD konumlarını örneğin, ilk G kodunu aynı blokta çıktısını almak için OUTGRD(1,i), i = 1, ..., 10 kullanılması mümkün değildir.

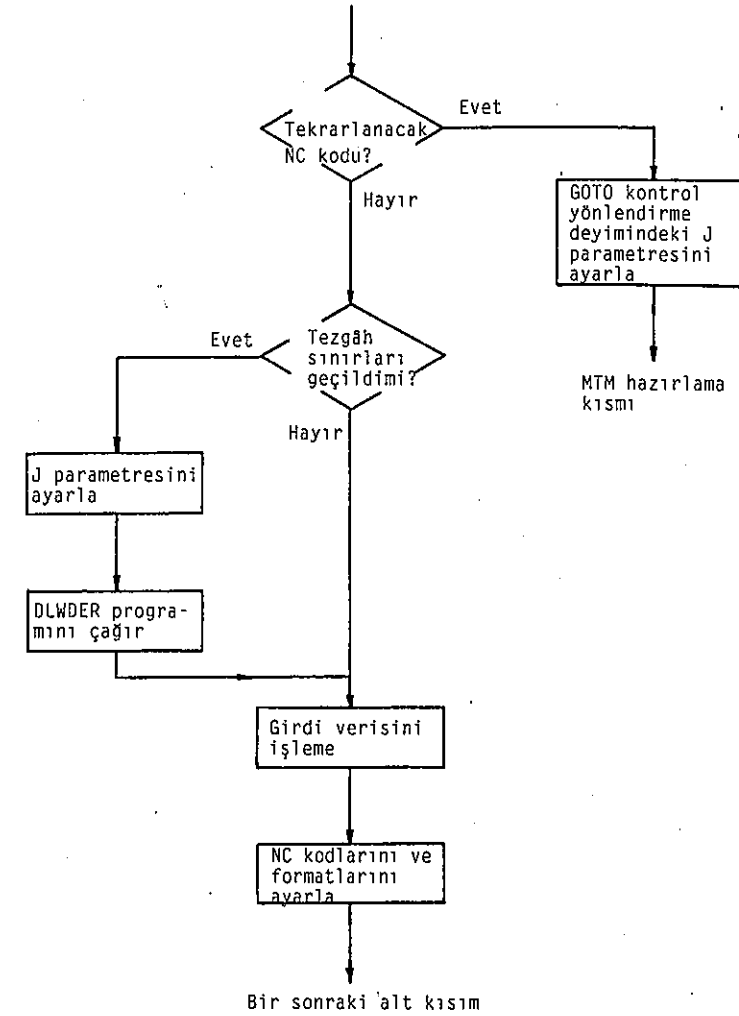
Görülebileceği üzere, NC'li tezgah için kullanılan kod sayısı 20'ye yakın olduğunda aynı kodu tanımlamak için iki veya daha fazla OUTGRD elemanının kullanımı, son işlemciyi gerekli bütün kodların işlenmesinden alıkoyabilir. Ayrıca normal olarak bir NC programda aynı cins kodları içeren NC deyimlerinin (bloklarının) sadece küçük bir oranı vardır. Onun için yukarıda verilen ikinci kuralın uygulanması NC programın saklanması için gerekli hafıza haciminde önemli bir artışa neden olur. Bu arada bu, MTM'nin tasarımını bazen kolaylaştırır.

Kural 2'ye göre, her işleme program bölümü daha evvelki kısımda işlenen bir kodun bu program kısmında işlenecek olan ile aynı tip olup olmadığını öncelikle tespit etmelidir. Eğer böyle ise, program kısmı GOTO denetim yönlendirme deyimindeki J parametresini halihazırda kısımda işaret eden bir değere ayarlar. Daha sonra denetimi daha önce işlenen kodun çıktısını vermek için MTM'nin hazırlama bölümüne gönderir. Kodun çıktısını verdikten sonra denetim MTM'ye geri döner ve GOTO denetim yönlendirme deyimini, denetimi, işleme devam etmesi için programa geri gönderir.

14.3.2 İşleme Özellikli Kodların İşlenmesi

İşleme özellikleri (örneğin, soğutucu, kesici, devir sayısı kodları) ile ilgili verileri işleyen alt bölümün işleme mantığı şöyle olabilir: İşlem başlamadan önce program önceki bölümde işlenen bir kodun bu bölümde işlenecek kod ile aynı olup olmadığını tespit etmelidir. Eğer öyle ise, denetim Bölüm 14.3.1'de açıklanan yönlendirilir. Eğer tekrarlanan kod yok ise, aşağıdaki işlem başlar.

Okunan veriler, işlenmiş verinin izin verilen aralıkta olup olmadığından emin olmak için önce tezgah sınırları ile karşılaştırılır. Tezgah sınırı aşılmışsa DLWDER programı hata mesajı yazmak için çağrılmalı ve sonra son işlemi bitirme ve devam etme gibi ilgili iş, uygun olarak yapılmalıdır. Aksi takdirde, işlem dev



Şekil 14-4 Tezgah operasyon kodunu işleyen MTM işleme alt bölümünün genel yapısı. Gerekirse, bir son işlemci tasarımcısı hata mesajı verildikten sonra son işlemi durdurmayı seçebilir.

eder. Okunan veri daha sonra DAPP çıktı bölümü tarafından doğrudan çıktısı verilen kodlara dönüştürülmelidir. Verileri çıktı için hazırlanan kodlara dönüştürüldükten sonra, kontrol bir sonraki işleme programına gider. Yukarıda açıklanan prensiplere dayalı olarak tasarlanan bir alt bölümün genel yapısı Şekil 14-4'de gösterilmektedir.

İş mili dönüş hızı ve yönünü ele alan alt bölümün, işleme operasyonunu açıklayan bir örnek olduğunu varsayalım. İş mili hızı, APT ve CLDATA'da olduğu

gibi dakikada devir sayısı olarak tanımlanır. İş mili hareketi ile ilgili bilgiyi içeren DATBUF elemanları DATBUF(13) ve (14)'tür. Eğer SD'li tezgah iş mili hızı ig dev./dak.'dan başka bir kod kullanırsa, DATBUF(14) uygun koda dönüştürülmelidir. Böylece uygun S kodu türetilir. Program kısmı M41 veya M42 kodunun aralığını da (örneğin Cadillac NC-100 tornası için) üretmelidir. İş mili hareketi çalıştırılmamışsa, program DATBUF(13) değerine dayalı olarak iş mili dönüş kodu M03 veya M04'ü üretmelidir. Aksi takdirde, program iş mili dönüşünün önce tanımlanan ile aynı yönde olup olmadığını tespit etmelidir. İki yönde aynı ise, M kodu türetilmemelidir; eğer iki dönüş yönü farklı ise bir M05 ve bir M04 veya M04 kodu dönüş yönünü tersine çevirmek için türetilir.

14.3.3 Kesici Hareketini Tanımlayan Kodların İşlenmesi

Basit bir kesici hareketi, kesicinin yolu, bu yol boyunca hareket hızı ve varış noktası ile tanımlanır. Çünkü, hareketin başlangıç noktası önceki hareketin son noktası olarak bilinir. Hareket kodlarını işleyen alt bölümün fonksiyonu DATBUF dizisinde saklanan bilgiyi freze tezgahı için G, X, Y, Z ve F veya torna için G, X, Z, U, W ve F doğru hareket kodlarına çevirmektir. Çevrimsiz kesici hareketini tanımlayan DATBUF elemanları şunlardır:

DATBUF(i) $i = 1-12, 19, 25$ (noktadan-noktaya ve doğrusal interpolasyon hareketi için)
 $i = 1-10, 36-71$ (daireysel interpolasyon hareketi için)

APT çevresel hareket deyimi kesici uç yarıçapı telafisini TLLFT, TLRGT, TLOFFPS veya TLOFPPS kelimelerinin kullanımı aracılığı ile hesaba kattığı için, normal olarak CLDATA'da kesici konumu kesici uç yarıçapı için telafi edilmesine gerek yoktur. O zaman DATBUF(27) daireysel interpolasyon düzlemini (yani, G17, G18 veya G19) tanımlamak için kod üretmek maksadı ile kullanılabilir. Çevrim kesici hareketi esnasında, sadece hızlı olmayan ilerlemeli noktadan-noktaya hareketi tanımlanabilir. Bunun için kesici hareketini tanımlayan konumlar:

DATBUF(i) $i = 1-11, 19, 25, 28-33$

Bundan dolayı, bir çevrimdeki geri çekilme hareketi, hızlı hareket olmalıdır. Çevrim fonksiyonu bulunmayan NC'li tezgahlar için ilgili G00 kodu türetilmelidir. Tablo 12-1'den görüleceği gibi kesici ekseninin uzaysal yönlendirilmesinin kaybolması için DATBUF dizisinde eleman yoktur. Böylece, eğer bir son işlemci çok eksenli fonksiyon işlemi içinde tasarlanırsa kesici eksen verisi INBUF dizisinden alınmalıdır (Bölüm 13.3'e bakınız).

14.3.31 Doğrusal-Enterpolasyon-Hareket Alt Bölümü. Doğrusal veya noktadan-noktaya hareketi tanımlayan veriler, ana fonksiyonu daha önce gösterilen DATBUF elemanlarında saklanan verileri G, X, Y, Z ve F doğru kodlarına çevirmek olan doğrusal enterpolasyon hareket alt bölümü (alt programı) tarafından işlenir. Ayrıca, tezgah dinamiğinden ortaya çıkan ihtiyaçların karşılanabilmesi için, fonksiyonlar bu kodları dönüştürmek veya işlemek için dahil edilmelidir.

NC denetleyici tezgahı iki farklı modda işletebilir: her hareket bloğunu farklı bir operasyon olarak icra edebilir veya hareket bilgisini bir bloktan bir sonrakine düzgün bir şekilde ve devamlı olarak enterpolasyon yapabilir. Böylece kesici hareketi esnasında gerçek sürekli çevresel hareket verir.

İlk kontrol metodu, sadece doğrusal kesme hareketini yapabilen NC'li matkap tezgahları veya eski tasarım NC denetleyicili NC'li tezgahlar gibi sadece noktadan-noktaya hareketli NC'li tezgahlar tarafından kullanılır. Bunların denetim sistemi, her hareket kademesinin başlangıcında programlanan ilerleme hızına kadar otomatik hızlanmayı ve sonunda duruncaya kadar yavaşlamayı temin etmek için tasarlanmıştır. Bir hareket bloğuyla tanımlanan hareket, önceki veya sonraki hareket bloğu tarafından tanımlanan blok tarafından etkilenmez. Bu durumda, genellikle tezgah dinamik probleminin son işlemci tasarımı esnasında düşünülmesine gerek yoktur⁸. İkinci tip denetim, sürekli çevresel hareketi icra eden NC denetim sistemlerinde uygulanır. Freze ve torna tezgahlarının NC denetleyicilerinin bir çoğu bu tiptedir. Bu tip denetim için durum farklıdır ve tezgah dinamik problemi uygun olarak göz önüne alınmalıdır.

Örneğin, APT'de tanımlanan hareketin, Şekil 14-5'de gösterildiği gibi, P0-B-C-D olduğunu varsayalım. Bu harekete karşılık gelen CLDATA, bir dizi kesici konum koordinatlarından ibarettir. $x_{P0}, y_{P0}, z_{P0}, x_b, y_b, z_b, x_c, y_c, z_c, x_{C1}, y_{C1}, z_{C1}, \dots, x_d, y_d, z_d$; C-D eğrisel hareketi, tanımlanan toleransa göre doğrusal enterpolasyon aracılığı ile NC işlemci tarafından birçok küçük doğrusal hareket parçalarına, C-C1, C1-C2, ..., bölünmüştür. Kesici, hareket yönünü aniden değiştirdiğinde tezgah dinamik problemleri ortaya çıkar. Örneğin, kesici P0 noktasından B ve C noktalarına hareket ederse, B noktasında fazla gitme oluşur. Çünkü hareketi azaltmak zaman alır. Kesicinin tanımlanan profilden sapması, S2, hem ileri gitme, S1 ve hem de hareket yönünde açılma değişmeye, A, bağlıdır. Böylece

$$S_1 = S_2 / \sin A$$

elde edilir. Burada A, iki hareket vektörü arasındaki açıdır (Şekil 14-5). Bu formülde S_2 tanımlı toleransı yerine koyarak, izin verilen S_1 fazla gitme elde edilebilir. Eğer A küçükse, S_2 sapması ihmal edilebilir. Böylece, C-C1-C2-...-D hareketi için fazla gitmenin profilin hassasiyeti üzerinde nispeten az bir etkisi vardır. Fazla gitmeyi azaltmak için, daha eskiden tasarlanan NC denetleyicisi,

gibi dakikada devir sayısı olarak tanımlanır. İş mili hareketi ile ilgili bilgiyi içeren DATBUF elemanları DATBUF(13) ve (14)'tür. Eğer SD'li tezgah iş mili hızı için dev./dak.'dan başka bir kod kullanırsa, DATBUF(14) uygun koda dönüştürülmelidir. Böylece uygun S kodu türetilir. Program kısmı M41 veya M42 kod hız aralığını da (örneğin Cadillac NC-100 tornası için) türetmelidir. İş mili henüz çalıştırılmamışsa, program DATBUF(13) değerine dayalı olarak iş mili doğru dönüş kodu M03 veya M04'ü türetmelidir. Aksi takdirde, program iş mili dönüşünün önce tanımlanan ile aynı yönde olup olmadığını tespit etmelidir. İki yönde aynı ise, M kodu türetilmemelidir; eğer iki dönüş yönü farklı ise bir M05 ve bir M03 veya M04 kodu dönüş yönünü tersine çevirmek için türetilir.

14.3.3 Kesici Hareketini Tanımlayan Kodların İşlenmesi

Basit bir kesici hareketi, kesicinin yolu, bu yol boyunca hareket hızı ve varsayılan noktası ile tanımlanır. Çünkü, hareketin başlangıç noktası önceki hareketin son noktası olarak bilinir. Hareket kodlarını işleyen alt bölümün fonksiyonu DATBUF dizisinde saklanan bilgiyi freze tezgahı için G, X, Y, Z ve F veya torna için G, X, Z, U, W ve F doğru hareket kodlarına çevirmektedir. Çevrimsiz kesici hareketini tanımlayan DATBUF elemanları şunlardır:

DATBUF(i) $i = 1-12, 19, 25$ (noktadan-noktaya ve doğrusal enterpolasyon hareketi için)
 $i = 1-10, 36-71$ (daireysel enterpolasyon hareketi için)

APT çevresel hareket deyimini kesici uç yarıçapı telafisini TLLFT, TLRGT, TLON-PS veya TLOFPS kelimelerinin kullanımı aracılığı ile hesaba kattığı için, normal olarak CLDATA'da kesici konumu kesici uç yarıçapı için telafi edilmesine gerek yoktur. O zaman DATBUF(27) daireysel enterpolasyon düzlemini (yani, G17, G18 veya G19) tanımlamak için kod türetmek maksadı ile kullanılabilir. Çevrim kesici hareketi esnasında, sadece hızlı olmayan ilerlemeli noktadan-noktaya hareketi ile tanımlanabilir. Bunun için kesici hareketini tanımlayan konumlar:

DATBUF(i) $i = 1-11, 19, 25, 28-33$

Bundan dolayı, bir çevrimdeki geri çekilme hareketi, hızlı hareket olmalı ve çevrim fonksiyonu bulunmayan NC'li tezgahlar için ilgili G00 kodu türetilmelidir. Tablo 12-1'den görüleceği gibi kesici ekseninin uzaysal yönlendirilmesinin kaydı için DATBUF dizisinde eleman yoktur. Böylece, eğer bir son işlemci çok eksenli fonksiyon işlemi içinde tasarlanırsa kesici eksen verisi INBUF dizisinden alınmalıdır (Bölüm 13.3'e bakınız).

14.3.31 Doğrusal-Enterpolasyon-Hareket Alt Bölümü. Doğrusal veya noktadan-noktaya hareketi tanımlayan veriler, ana fonksiyonu daha önce gösterilen DATBUF elemanlarında saklanan verileri G, X, Y, Z ve F doğru kodlarına çevirmek olan doğrusal enterpolasyon hareket alt bölümü (alt programı) tarafından işlenir. Ayrıca, tezgah dinamiğinden ortaya çıkan ihtiyaçların karşılanabilmesi için, fonksiyonlar bu kodları dönüştürmek veya işlemek için dahil edilmelidir.

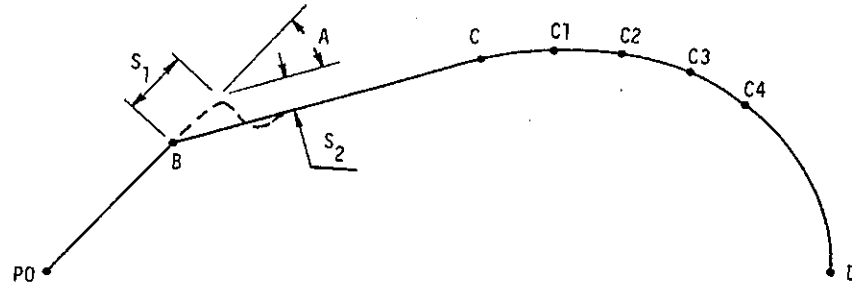
NC denetleyici tezgahı iki farklı modda işletebilir: her hareket bloğunu farklı bir operasyon olarak icra edebilir veya hareket bilgisini bir bloktan bir sonrakine düzgün bir şekilde ve devamlı olarak enterpolasyon yapabilir. Böylece kesici hareketi esnasında gerçek sürekli çevresel hareket verir.

İlk kontrol metodu, sadece doğrusal kesme hareketini yapabilen NC'li matkap tezgahları veya eski tasarım NC denetleyicili NC'li tezgahlar gibi sadece noktadan-noktaya hareketli NC'li tezgahlar tarafından kullanılır. Bunların denetim sistemi, her hareket kademesinin başlangıcında programlanan ilerleme hızına kadar otomatik hızlanmayı ve sonunda duruncaya kadar yavaşlamayı temin etmek için tasarlanır. Bir hareket bloğuyla tanımlanan hareket, önceki veya sonraki hareket bloğu tarafından tanımlanan blok tarafından etkilenmez. Bu durumda, genellikle tezgah dinamik probleminin son işlemci tasarımı esnasında düşünülmesine gerek yoktur⁸. İkinci tip denetim, sürekli çevresel hareketi icra eden NC denetim sistemlerinde uyarlanır. Freze ve torna tezgahlarının NC denetleyicilerinin bir çoğu bu tiptedir. Bu tip denetim için durum farklıdır ve tezgah dinamik problemi uygun olarak göz önüne alınmalıdır.

Örneğin, APT'de tanımlanan hareketin, Şekil 14-5'de gösterildiği gibi, P0-B-C-D olduğunu varsayalım. Bu harekete karşılık gelen CLDATA, bir dizi kesici konum koordinatlarından ibarettir. $x_{P0}, y_{P0}, z_{P0}, x_b, y_b, z_b, x_c, y_c, z_c, x_{C1}, y_{C1}, z_{C1}, \dots, x_d, y_d, z_d$; C-D eğrisel hareketi, tanımlanan toleransa göre doğrusal enterpolasyon aracılığı ile NC işlemci tarafından birçok küçük doğrusal hareket parçalarına, C-C1, C1-C2, ..., bölünmüştür. Kesici, hareket yönünü aniden değiştirdiğinde tezgah dinamik problemleri ortaya çıkar. Örneğin, kesici P0 noktasından B ve C noktalarına hareket ederse, B noktasında fazla gitme oluşur. Çünkü hareketi azaltmak zaman alır. Kesicinin tanımlanan profilden sapması, S2, hem ileri gitme, S1 ve hem de hareket yönünde açısal değişmeye, A, bağlıdır. Böylece

$$S_1 = S_2 / \sin A$$

elde edilir. Burada A, iki hareket vektörü arasındaki açıdır (Şekil 14-5). Bu formülde S_2 tanımlı toleransı yerine koyarak, izin verilen S_1 fazla gitme elde edilebilir. Eğer A küçükse, S_2 sapması ihmal edilebilir. Böylece, C-C1-C2-...-D hareketi için fazla gitmenin profilin hassasiyeti üzerinde nispeten az bir etkisi vardır. Fazla gitmeyi azaltmak için, daha eskiden tasarlanan NC denetleyicisi,



Şekil 14-5 APT'de P0-B-C-D olarak tanımlanan kesici hareketi. CLDATA'da P0-B-C-C1-C2-C3-...-D olarak yeniden tanımlanır.

S_1 : Fazla gitme; S_2 : Fazla gitmenin sebep olduğu sapma;

----- : Köşedeki gerçek kesici yolu;

— : Programlanan kesici yolu.

kesici varış noktasına yakın olduğunda (yani B noktası) her hareketin (P0-B gibi) yavaşlatılması gerekir. Böylece, eğer programlanan ilerleme hızı, aşırı fazla gitmeye neden oluyorsa, her hareket kademesinin ikiye bölünmesi gerekir; birincisi programlanan ilerleme ve diğeri daha az ilerleme. Genellikle, ilerleme-fazla gitme ilişkisi ile ilgili veriler ve ilerleme ve en az hızlanma veya yavaşlama kademesi arasındaki ilişki (Tablo 14-1[a ve b]) takım tezgahı üreticisi tarafından hazırlanır.

TABLO 14-1(a) PROGRAMLANAN İLERLEME HIZININ BİR FONKSİYONU OLARAK FAZLA GİTME S_1 (ŞEKİL 14-5'E BAKINIZ)

İlerleme (inç/dak.)	Fazla gitme S_1 (inç)
4.0	0
10.0	0.0004
20.0	0.0008
40.0	0.0020

TABLO 14-1(b) PROGRAMLANAN İLERLEMeye HIZLANDIRMAK VEYA PROGRAMLANAN İLERLEMEDEN SIFIRA YAVAŞLATMAK İÇİN GEREKLİ EN AZ MESAFE

İlerleme (inç/dak.)	En az Mesafe (inç)
10	0.02
20	0.02
40	0.04
80	0.08

İzin verilen sapma, DATBUF(26)'da saklanan aşağıdaki deyim kullanılarak bir APT programcısı tarafından tanımlanır:

MCHTOL/n

Şimdi kesici hareketini işlemek için algoritma tasarlanabilir.

Modern BSD tezgahlarında bu durum farklıdır. Hemen hemen bütün BSD freze ve torna tezgahları otomatik hızlanma ve yavaşlama (a/d) fonksiyonuna sahiptirler ve üstel a/d zaman sabiti, iç köşedeki a/d mesafeleri ve kesme hareketinde en çok konum sapması gibi karakteristik parametreler kullanıcının ihtiyacına göre ayarlanabilir. Bundan dolayı, otomatik a/d fonksiyonlu CNC tezgahları için bir doğrusal kesme hareketi sonunda yavaşlamayı sağlayan bir program kısmının eklenmesine gerek yoktur. Çok yüksek bir ilerleme kullanıldığında, hala aşırı fazla gitmeye neden olunabilir. Fakat, bu sadece kaba talaş esnasında olur ve normal ilerleme ile bitirme işleminden sonra parça profilinin tolerans aralığında tutulabilmesi için yeterli izin verilen pay yüzeyde bırakılmalıdır.

Bazı eski tasarımlı NC denetleyiciler farklı gitme mesafeli doğrusal enterpolasyon hareketini tanımlamak için farklı G kodlarının kullanımını gerektirirler. Ayrıca bu kodlar için denetleyicinin hassasiyeti de (en az artışlı hareket mesafesi) farklıdır. Örneğin, Bunker-Ramo 3100 denetleyicili 10 Gorton Tapemaster 2-30 çevresel freze X, Y ve Z kodlarının ondalık noktası olmaksızın 1 ila 4 arası rakamlar olarak girilmelerini gerektirir. Eğer G01 kodu etkin ise ondalık noktasının tanımlanan ilk rakamdan önce geleceği veya G10 kodu ile birinci ve G60 kodu ile ikinci rakamdan sonra konacağı farz edilmektedir. G01, G10 ve G60 kodları ile programlanabilen X, Y ve Z kodları artışlı değerler olmalıdır. G01, G10 ve G60 kodları için en az programlanabilen hareket mesafeleri veya hassasiyetleri sırası ile 0.0001, 0.001 ve 0.01'dir; diğer yandan bu üç kodlu en çok ayrı ayrı kalkış mesafeleri 0.9999, 9.999 ve 99.99'dur. Onun için, X eksenı boyunca 14.2345 artışlı bir hareket G01, G10 veya G60 bir hazırlık fonksiyon kodu ile programlanamaz. Üç hareket kademesine bölünmelidir; örneğin

$$14.2345 = 12.94 + 1.177 + 0.1175$$

her biri uygun bir G kodu ile tanımlanır. Eğer hareket X yönünde ise, üç blok gerekir:

G60X12.94

G10X1.177

G01X0.1175

Aynı anda üç eksen hareketi için, hareketin doğrusallığının korunabilmesi için başlangıç noktasından X, Y ve Z eksenlerindeki son noktaya olan artışlı mesafeler

orantılı bir şekilde parçalara ayrılmalıdır. Son işlemci tasarımcısına doğrusal hareketin parçalara ayrılmasında DAPP'a eklenen iki özel program bölümü DLWSRR ve DLWSEG tarafından yardım edilir. Bu iki program SEGCOM etiketli COMMON bölgesinde tanımlanan değişkenler DAPP ve MTM ile iletişim için kullanılırlar:

```
COMMON/SEGCOM/RANGE1,RESLN1,RANGE2,RESLN2,RANGE3,RESLN3,NRANGE
```

RANG1 ve RESLNE1, en çok artışlı mesafeyi ve en büyük tanımlanabilen artışlı hareket kademeli G kodu için olan hassasiyeti (örneğin, yukarıda gösterilen örnekteki G60) tanımlamak için kullanılırlar; RANGE3 ve RESLN3, en az tanımlanabilen artışlı hareket kademeli G kodu için (örneğin, yukarıdaki örnekte gösterilen G01) benzer şekilde kullanılır. NRANGE değişkeni doğrusal enterpolasyon hareketinin tanımlanması için kullanılan G kodlarının sayısını tanımlamak için kullanılır. Yukarıda gösterilen COMMON deyimi MTM'nin değişken tanımlama kısmında tanımlanmalıdır ve bu değişkenler bu iki program bölümü çağrılmadan önce tanımlanmalıdır. Yukarıda verilen örnek için, bunlar aşağıdaki gibi tanımlanabilirler:

```
NRANGE=3
RANGE1=99.99D0
RANGE2=9.999D0
RANGE3=0.9999D0
RESLN1=0.01D0
RESLN2=0.001D0
RESLN3=0.0001D0
```

Bunlar, DAPP'ta bu değişkenler için de ayrı ayrı varsayılan değerler olarak kullanılırlar. Aralıklar ve hassasiyetler büyükten küçüğe doğru ayarlanıp düzenlenmelidirler. Örneğin, eğer bir NC'li tezgah doğrusal enterpolasyon hareketini tanımlamak için sadece iki G kodu, örneğin G01 ve G10, kullanılsa, bu değişkenler aşağıdaki gibi tanımlanmalıdır:

```
NRANGE=2
RANGE1=9.999D0
RANGE2=0.9999D0
RESLN1=0.001D0
RESLN2=0.0001D0
```

Bu FORTRAN deyimleri MTM'nin ilk değerlerine atama bölümüne eklenebilir. DLWSRR yordamının fonksiyonu, yukarıda açıklandığı gibi tanımlanan

NRANGE, RANGE1, ..., RESLN3 değişkenlerine göre en çok parça uzunluklarına ve hassasiyetlere ayarlamak içindir. Kesici yolunun parçalara ayrılmasını yapan DLWSEG yordamını çağırılmadan önce, aşağıdaki deyim kullanılması ile çağrılabilir:

```
CALL DLWSRR
```

Aşağıdaki deyim DLWSEG program kısmını çağırır:

```
CALL DLWSEG (DATBUF(7),M,OUTPUT,IRET)
```

burada

- DATBUF(7) = Daha sonra parçalara ayırma işlemi icra eden DLWSEG yordamı tarafından kullanılan AINPUT dizisinde saklanacak artışlı hareketi tanımlayan üç elemana, DATBUF(7) ile (9) arası, neden olan bir parametredir.
- M = OUTPUT dizisinde olan elemanların sayısını tanımlayan tam sayı.
- OUTPUT = Parçalara ayrılmış verilerin saklandığı M elemanlı dizinin adı (son işlemci tasarımcısı tarafından verilen bir isim ile değiştirilebilir, örneğin, Şekil 146'da verilen akış diyagramındaki D).
- IRET = OUTPUT dizisinde, mutlak değerinin DLWSEG yordamı tarafından doldurulan eleman sayısını temsil eden dönüş kodu (Eğer AINPUT(1) ile (3) arasındaki dizide veriler tamamen parçalara ayrılmış ve sıfıra indirgenmişse dönüş kodu pozitifdir. Eğer M parametresi tarafından atanan OUTPUT dizisinin konumlarının sayısı gerekenden az ise IRET dönüş kodu negatiftir).

Bir örnek olarak Gorton Tapemaster 2-30 çevresel işleme frezesini tekrar ele alalım. Bir kesici hareketi X, Y ve Z değerlerinin sırası ile 27.2543, 6.2543 ve 0.2010'a eşit artışlı programlanmıştır. Bunlar sırası ile DATBUF(7) ile (9) arasında saklanmaktadır. DLWSEG yordamı aşağıdaki deyim tarafından çağrılır:

```
CALL DLWSEG (DATBUF(7),9,OUTPUT,IRET)
```

Sonuç olarak, hareket üç parçaya bölünür ve aşağıdaki veriler türetilir:

OUTPUT(1) = 24.78	} (G60 kodu için artışlı X, Y ve Z değerleri)
OUTPUT(2) = 5.69	
OUTPUT(3) = 0.18	
OUTPUT(4) = 2.249	} (G10 kodu için artışlı X, Y ve Z değerleri)
OUTPUT(5) = 0.513	
OUTPUT(6) = 0.019	
OUTPUT(7) = 0.2253	} (G01 kodu için artışlı X, Y ve Z değerleri)
OUTPUT(8) = 0.0513	
OUTPUT(9) = 0.0020	

Eğer çağırma deyiminde M parametresi 3'e ayarlanırsa, DLWSEG yordamı, hareketi tamamen parçalara bölmek için üç kere çağrılmalıdır. DLWSEG yordamının her çağrılışında türetilen veriler Tablo 14-2'de verilmektedir.

TABLO 14-2 DLWSEG YORDAMI ÇAĞRILDIKTAN SONRA ORTAYA ÇIKAN VERİLER

	CALL DLWSEG (DATBUF (7),3,OUTPUT,IRET)		
	Birinci Çağırma	İkinci Çağırma	Üçüncü Çağırma
OUTPUT (1)	24.78	2.249	0.2253
OUTPUT (2)	5.69	0.513	0.0513
OUTPUT (3)	0.18	0.019	0.0020
IRET	-3	-3	+3
DATBUF (7)*	2.4743	0.2253	0
DATBUF (8)*	0.5643	0.0513	0
DATBUF (9)*	0.0210	0.0020	0

*Sırası ile bu üç değişkenlerin ilk değerleri 27.2543, 6.2543 ve 0.2010'dur.

Bu veriler, G60, G10 ve G01 kodları ile tanımlanan doğrusal enterpolasyon hareket parçaları için X, Y ve Z kodlarını üretmek maksadı ile kullanılabilir. Bu örnekten görülebileceği üzere, her veri elemanı, örneğin OUTPUT(1), (2) ve (3), DATBUF elemanındaki veriyi 1.1 ile bölmekle türetilir ve son parçanın değeri, örneğin OUTPUT(7), programlanan mesafe, DATBUF(7) ile önceden türetilen ilgili iki parçanın, OUTPUT(1) ve OUTPUT(4), toplamı arasındaki farktır. Hesaplama yuvarlatıldığı için, özellikle farklı hareket parçalarındaki artışlı X, Y ve Z değerleri arasındaki oranlar aynı tutulamayabilir,

$$\begin{aligned} & \text{OUTPUT}(1) : \text{OUTPUT}(2) : \text{OUTPUT}(3) \\ & \neq \text{OUTPUT}(4) : \text{OUTPUT}(5) : \text{OUTPUT}(6) \\ & \neq \text{OUTPUT}(7) : \text{OUTPUT}(8) : \text{OUTPUT}(9) \end{aligned}$$

Böylece sonuçta ortaya çıkan kesici yolu tam olarak bir doğru değildir. OUTPUT

dizisinin ilk üç elemanının sırası ile X, Y ve Z kodları olarak G60 kodu ile çıktısı verilir; sonraki iki deyim G10 ve G01 kodlu X, Y ve Z kodlarıdır. X, Y ve Z kodlarının formatı G60, G10 ve G01 kodları için aynı olduğundan dolayı (ondalık noktasız dört rakam) sadece OUTGRD elemanlarının üç grubunu kullanabiliriz; örneğin X, Y ve Z kodlarının formatlarını tanımlamak için:

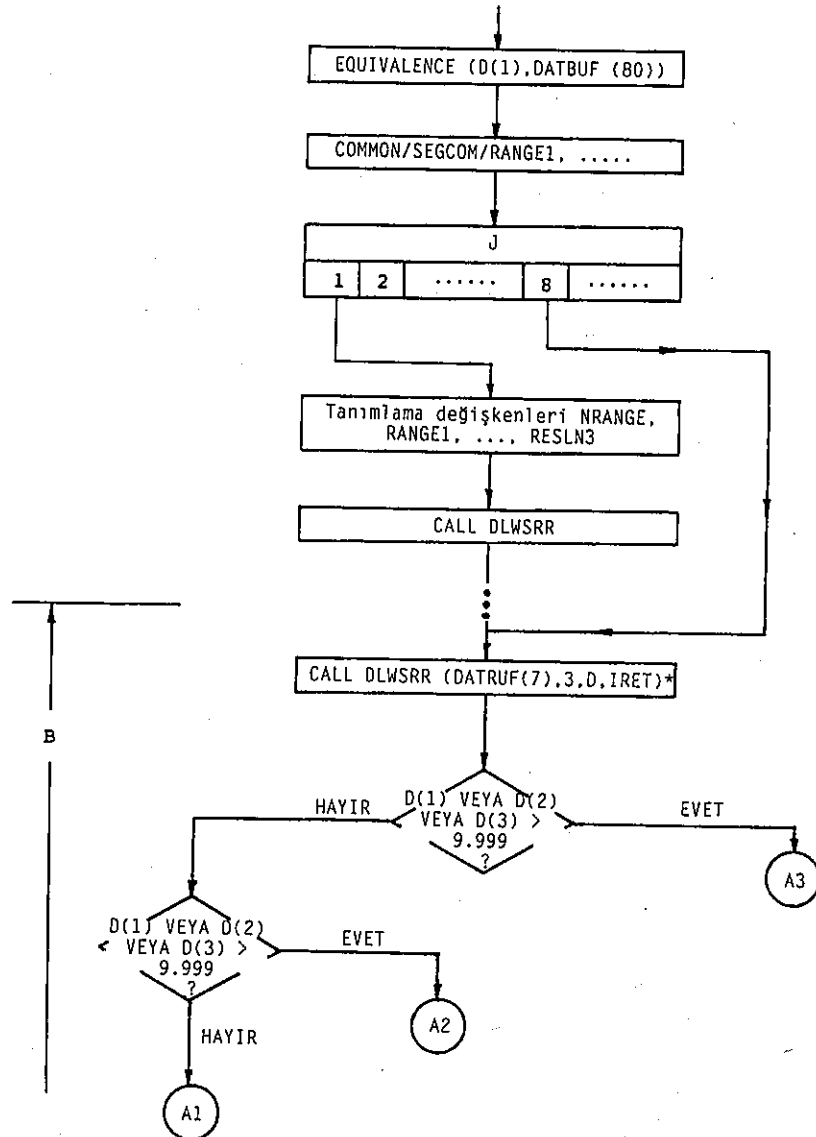
$$\text{OUTGRD}(i,j) \quad i = 3, 4, 5 \text{ ve } j = 1, \dots, 9$$

Program Şekil 14-6'da gösterildiği gibi tasarlanabilir. SEGCOM etiketli COMMON alanı ve değişkenleri tanımlandıktan sonra DLWSRR yordamı çağrılır. Sonra DATBUF(7) ile (9) arasında saklanan hareket verileri işlenebilir. Önce, içinde M parametresinin 3'e ayarlandığı DLWSEG yordamı hareketi parçalara bölmek için çağrılır. Yukarıdaki açıklamalardan da görüleceği üzere, hareketin parçalara ayrılması aşağıdaki prensibe göre DLWSEG tarafından icra edilir: Doğrusal bir hareket

- Eğer $\text{Max}(|\text{DATBUF}(7)|, |\text{DATBUF}(8)|, |\text{DATBUF}(9)|) > 9.999$ ise üç parçaya
- Eğer $0.9999 < \text{Max}(|\text{DATBUF}(7)|, |\text{DATBUF}(8)|, |\text{DATBUF}(9)|) \leq 9.999$ ise iki parçaya bölünür ve
- Eğer $\text{Max}(|\text{DATBUF}(7)|, |\text{DATBUF}(8)|, |\text{DATBUF}(9)|) \leq 0.9999$ ise bölmeye gerek yoktur.

Böylece programın icrası, maksimum hareket elemanlarının, D(1), D(2) veya D(3) değerine bağlı olarak üç farklı bölüme dallanır. Maksimum hareket elemanı G01, G10 ve G60 kodları ile tanımlanırsa sırası ile A1, A2 ve A3 bölümleri girilir. Her bölümde, önce işlem program kısmı önceki bloktaki G kodunun türetmesi gereken kodla aynı olup olmadığını tespit eder. Eğer öyle ise, yeni bir G01, G10 veya G60 kodunun türetilmesine gerek yoktur, aksi takdirde yordam henüz çıktısı verilmemiş, önceki bölümde işlenmiş olan G kodunun olup olmadığını inceler (aynı tipteki iki kodun bir blokta tanımlanamayacağı prensibi tatbik edilir). Eğer öyle ise, önceden işlenmiş olan G kodu önce çıktı olarak verilmelidir, aksi takdirde uygun X, Y, Z ve G kodları OUTPUT dizisine (Şekil 14-6'daki D dizisi) göre ayarlanır. Daha sonra ilerleme hızı ya programlanan ilerlemeye göre veya fazla gitmeyi telafi etmek için azaltılmış ilerlemeye göre ayarlanmalıdır.

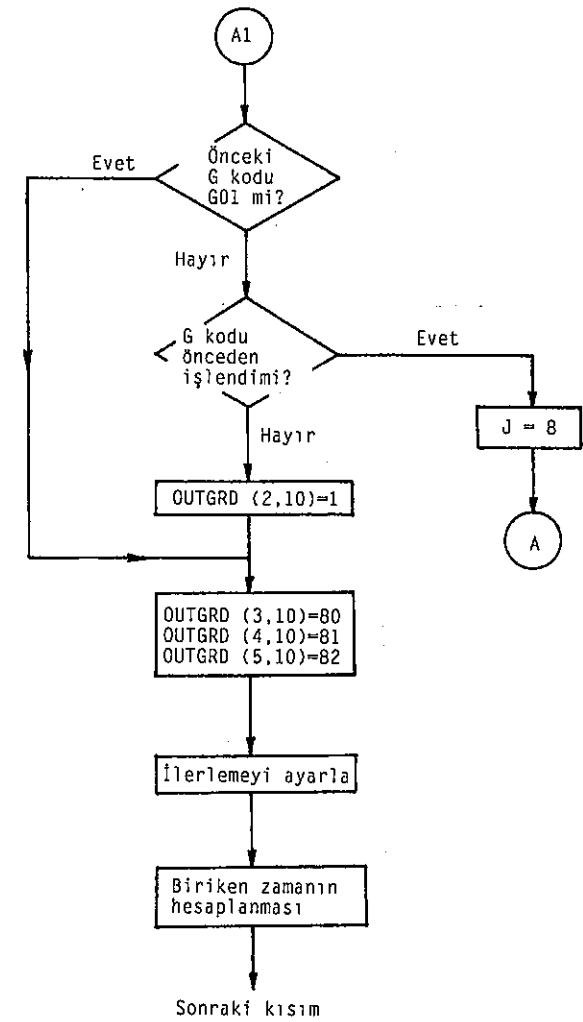
Modern CNC tezgahları için, doğrusal enterpolasyon hareketi tek bir kodla, G01, tanımlanabilir. DLWSRR ve DLWSEG yordamları kullanmaya gerek yoktur ve sadece bir işleme kısmı gerekir. Onun için doğrusal enterpolasyon hareketini



Şekil 14-6 Gorton Tapemaster 2-30 freze tezgahı için doğrusal enterpolasyon hareketini işleyen bölümün akış diyagramı. OUTGRD(2,i), OUTGRD(3,i), OUTGRD(4,i) ve OUTGRD(5,i) (i = 1, ..., 10) elemanları sırası ile G, X, Y ve Z kodları için kullanılır*: D sembolü OUTPUT dizisini adlandırmak için kullanılır. A: TTM'nin hazırlama bölümü; B: Doğrusal enterpolasyon hareketini işleyen bölüm.

İşleyen yordam büyük ölçüde basitleştirilebilir.

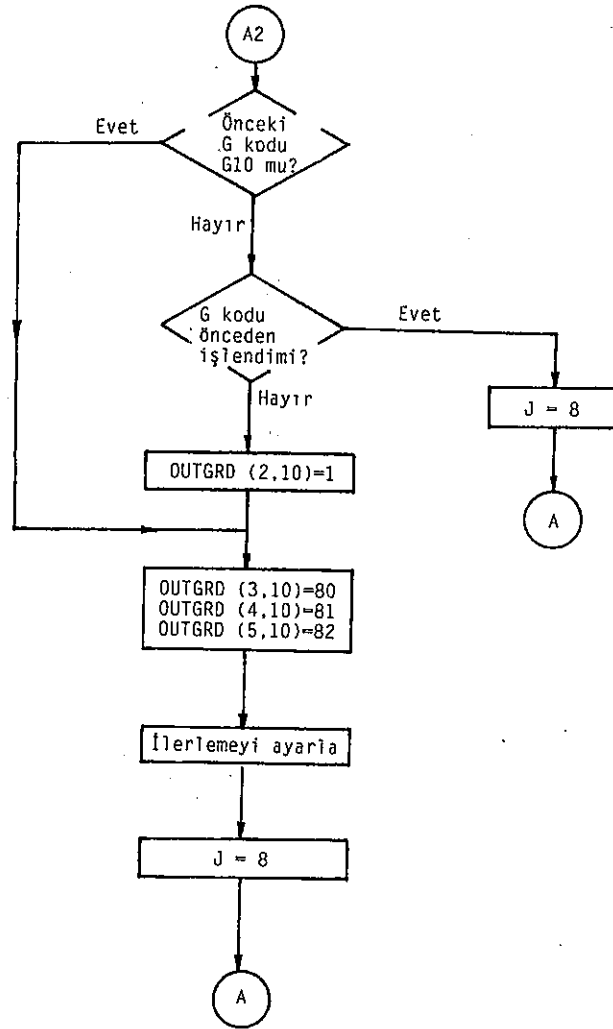
Hızlı noktadan-noktaya hareket, doğrusal enterpolasyon bölümü tarafından



Şekil 14-6 (devamı var)

işlenebilir. Çünkü bu iki hareketlerin tanımlanmasındaki fark sadece onların G kodlarındadır. G kodu programlanan ilerleme hızından tespit edilebilir. Böylece, G kodu ayarlanmadan önce ilerlemenin kontrol edilmesi gerekir. Eğer ilerleme RAPID (yani hızlı) ise, G kodu G00 olarak ayarlanmalıdır, aksi takdirde G01'e ayarlanmalıdır.

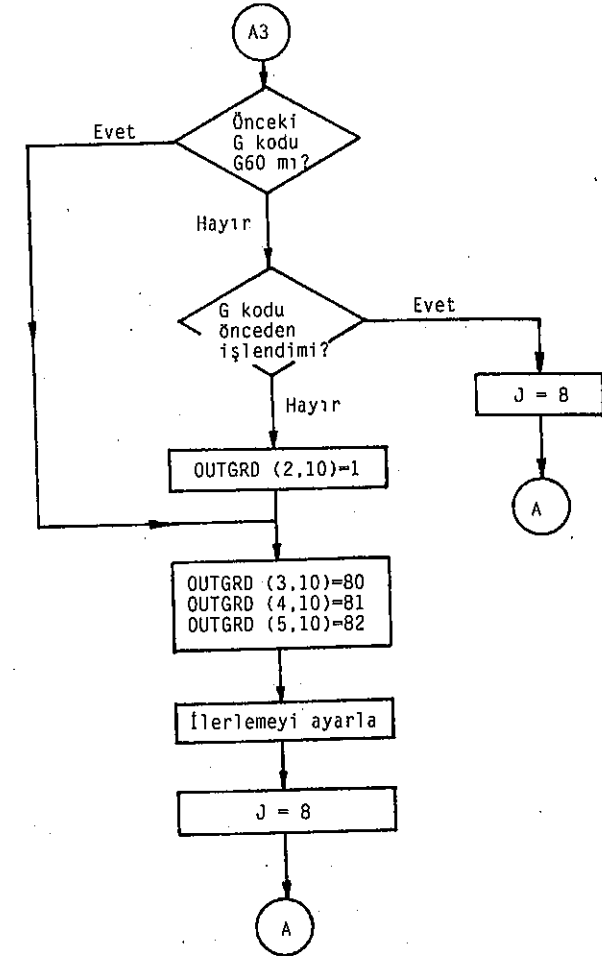
Bu yordamdaki son işleme safhası, halihazırdaki deyim kadar NC programın icrası için biriken zamanın hesaplanmasından ibarettir. Bu aşağıdaki FORTRAN deyiminin kullanımı ile gerçekleştirilir:



Şekil 14-6 (devamı var)

CLKARG=CLKARG+TIME(1)

Bu torna için tasarlanan son işlemciler için, DATBUF dizisindeki artışı x ve y koordinatları sırası ile W ve UNC kodlarına çevrilmelidir (Bölüm 8.1'e bakınız); Ayrıca, U kodu çap değeri olmalıdır (yani, DATBUF dizisindeki değerin iki katı). 15. Bölümde verilen örnek bu çevrilmeyi detaylı bir şekilde gösterir.



Şekil 14-6 (devamı var)

14.3.3.2 Dairesel Enterpolasyon Hareket Alt Bölümü Bölüm 13.4.2.2'de açıklandığı gibi, dairesel bir hareket DAPP'ın DLWCIR yordanı tarafından 1 ile 5 arasında parçalara bölünür ve sonuç verileri elemanlarda saklanır:

DATBUF(i) i = 36-71

Dairesel bir yolun parçalara ayrılması bazı eski tasarımlı NC'li tezgahlar için gerekir. Fakat, modern CNC denetimciler için gerekli değildir, FANUC 6MB ve 6T gibi.

Bu yordanın ana fonksiyonu, verileri Tablo 14-3'te verilen karşılıklara göre doğru SD kodlarına çevirmektir:

DATBUF(i) i = 1-10, 36-71

TABLO 14-3 DATBUF DEĞİŞKENLERİ VE ONLARA KARŞILIK GELEN KODLARI ARASINDAKİ İLİŞKİ

DATBUF Dizisindeki Elemanlar	İlgili SD Kodları
1-3	X, Y, Z (mutlak değer)
4-6	X (veya W*), Y, Z (veya U*) (artışlı değer)
10	F
36	G17, G18 veya G19
37	G02 veya G03
41	R
42-44, 48-50, 54-56	X (veya W*), Y, Z (veya U*) (artışlı değer)
60-62, 66-68	
45-47, 51-53, 57-59, 63-65, 69-71	I, J, K (aşağıdaki yazıya bakınız)

*Torna için U ve W kodları

Bölüm 13.4.2.2'de ifade edilen DLWCIR hesaplama yordamındaki hatalardan dolayı, merkez kaydırmaları için DATBUF konumları I, J ve K kodlarının çıktısını vermek için kullanılamazlar:

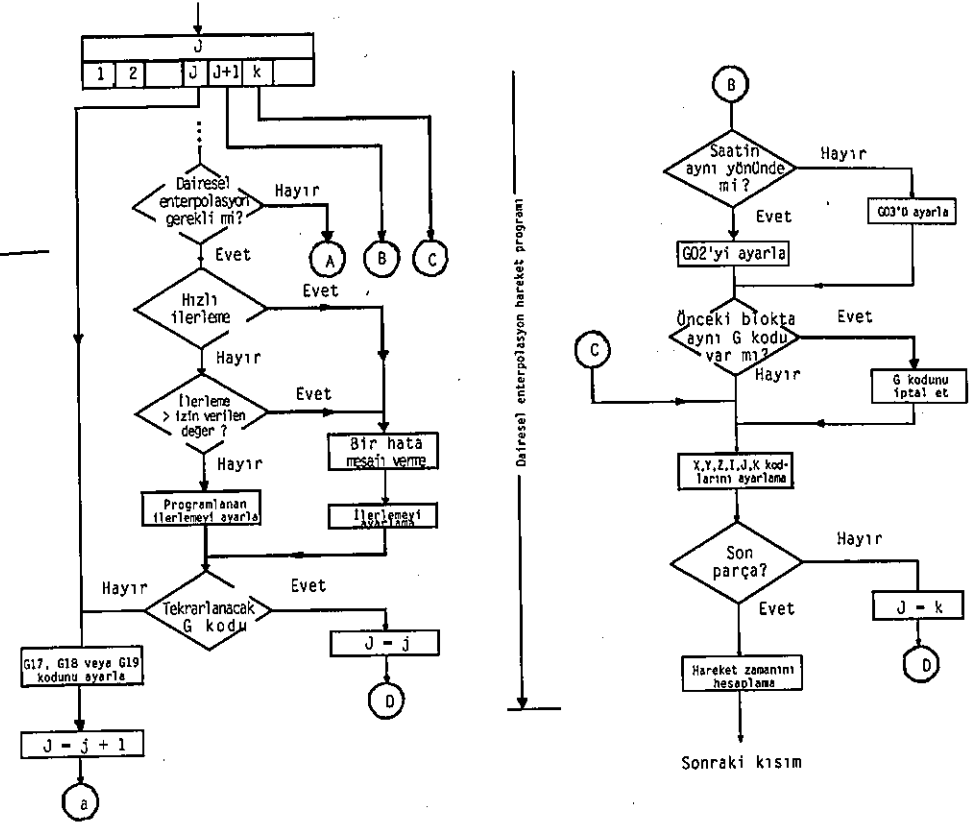
$$\text{DATBUF}(i) \quad i = 45-47, 51-53, 57-59, 63-65, 69-71$$

Bu değerler MTM'de aşağıdaki gibi yeniden hesaplanmalıdır:

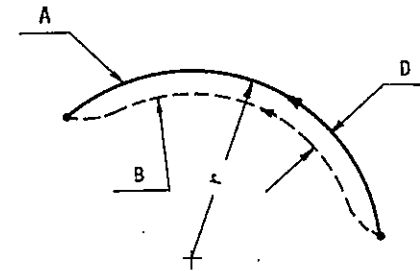
$$\begin{aligned} \text{DATBUF}(44 + i) &= \text{DATBUF}(37 + i) - \text{DATBUF}(3 + i) \\ \text{DATBUF}(50 + i) &= \text{DATBUF}(37 + i) - [\text{DATBUF}(3 + i) + \text{DATBUF}(41 + i)] \\ \text{DATBUF}(56 + i) &= \text{DATBUF}(37 + i) - [\text{DATBUF}(3 + i) + \text{DATBUF}(41 + i) + \text{DATBUF}(18)(47 + i)] \\ \text{DATBUF}(62 + i) &= \text{DATBUF}(37 + i) - [\text{DATBUF}(3 + i) + \text{DATBUF}(41 + i) + \text{DATBUF}(47 + i) + \text{DATBUF}(53 + i)] \\ \text{DATBUF}(68 + i) &= \text{DATBUF}(37 + i) - [\text{DATBUF}(3 + i) + \text{DATBUF}(41 + i) + \text{DATBUF}(47 + i) + \text{DATBUF}(53 + i) + \text{DATBUF}(59 + i)] \end{aligned}$$

burada $i = 1, 2$ ve 3 'tür.

Dairesel enterpolasyon hareket alt bölümünün genel yapısı Şekil 14-7'de gösterilmektedir. Dairesel enterpolasyon işleme ARCSLP/ON deyimi tarafından istendiğinde bu girilir. Sonra ilerleme hızının RAPID'den başkası olup olmadığından emin olmak için ilerleme hızı kontrol edilir. Bu yordam, programlanan ilerlemenin parça profilinden (Şekil 14-8) aşırı sapmaya neden olup olmayacağını da tespit edebilir. Dairesel enterpolasyon hareketi için izin verilen ilerleme, NC sisteminin özellikleri ve tanımlanan profilden izin verilen sapma (yani tolerans) tarafından tespit edilir. Genellikle dairesele enterpolasyon hareketi esnasında radyal yöndeki en büyük sapmanın hesaplanması için denklem, takım tezgahı yapımcısı



Şekil 14-7 Dairesel enterpolasyon hareket verilerini işleyen alt bölümün yapısı. A: Doğrusal enterpolasyon hareket kısmı; D: MTM'nin hazırlama bölümü.



Şekil 14-8 Denetim sisteminden dolayı, gerçek kesici yolunun (B) programlanan kesici yolundan sapması. D: Maksimum sapma.

veya NC denetimci üreticisi tarafından hazırlanır. Örneğin, FANUC 6MB denetimci için olan denklem*

$$D = [(T_1^2 + T_2^2) \cdot f^2] / (2 \cdot r)$$

*Bu Kitaptaki Kısım I için referans listesindeki 26. referansa bakınız.

burada

- D = Radyal yönde maksimum sapma (inç)
 T_1 = NC Denetimcinin üstsel hızlanma veya yavaşlama zaman sabiti (saniye, genellikle sıfıra ayarlanır)
 T_2 = Konum döngü kazancının tersi olan konumlama denetim sistemi için zaman sabiti (saniye)
 f = İlerleme (inç/saniye)
 r = Yay yarıçapı (inç)

Eğer

$$T_1 = 0$$

$$T_2 = 3.3 \times 10^{-2} \text{ saniye} \quad \text{dir.}$$

ve D D' toleransı (inç) ise, o zaman izin verilen ilerleme:

$$f = 2571.3 \cdot (D' \cdot r)^{-1/2} \text{ (inç/dak.)}$$

Tablo 14-4, farklı yarıçaplar ve bu formülle hesaplanan toleranslar için izin verilen ilerlemeleri listelemektedir. Hemen görülebileceği üzere, bitirme ve birçok kaba işleme operasyonları için izin verilen ilerleme aşılmamaktadır.

Bu yordamın bir sonraki işi, DATBUF dizisindeki verileri G, X, Y, Z ve F kodlarına çevirmektir. Bir dairesel hareket deyimi (bloğu) kesinlikle bir G kodu içereceği için (eğer varsa) önceki kısımda işlenen G kodunun, MTM'nin hazırlama bölümüne kontrolün dallandırılması ile çıktısı önce alınmalıdır. Aksi takdirde, dairesel enterpolasyon başlar. Önce program dairesel enterpolasyon düzlem kodunu DATBUF(36)'un değerine göre ayarlar. Tekrar kontrol, G17, G18 veya G19 kodunu çıktı olarak vermek için hazırlama bölümüne yönlendirilir (Çünkü aynı tipteki iki kod bir NC blokta tanımlanamayacağı prensibini takip ediyoruz). Sonra G02 veya G03 kodu DATBUF(37)'de tanımlanan hareket yönüne göre ayarlanır. Eğer dairesel hareket farklı çeyreklerde parçalara bölünmesi gerekiyorsa, o zaman (hareket X-Y düzleminde ise) X, Y, I ve J kodları ilk parça için ayarlanmalıdır. Kontrolü hazırlama bölümüne dallandırmakla bu kodların çıktıları alınır. Bu kodların çıktısı alındıktan sonra kontrol geri gelir ve bir sonraki hareket parçası için X, Y, I ve J kodlarını ayarlar. Bu işlem bütün hareket parçaları tanımlanana kadar tekrarlanır. Dairesel hareketin parçalanmasının gerekmediği NC'li tezgahlar için aşağıdaki formül tarafından verilen bitiş koordinatları X, Y ve Z kodlarını işlemek için kullanılabilir:

$$\text{DATBUF}(i) \quad i = 1, 2, 3$$

TABLO 14-4 FARKLI TOLERANSLAR, D' (inç) VE YARIÇAPLAR, r (inç)

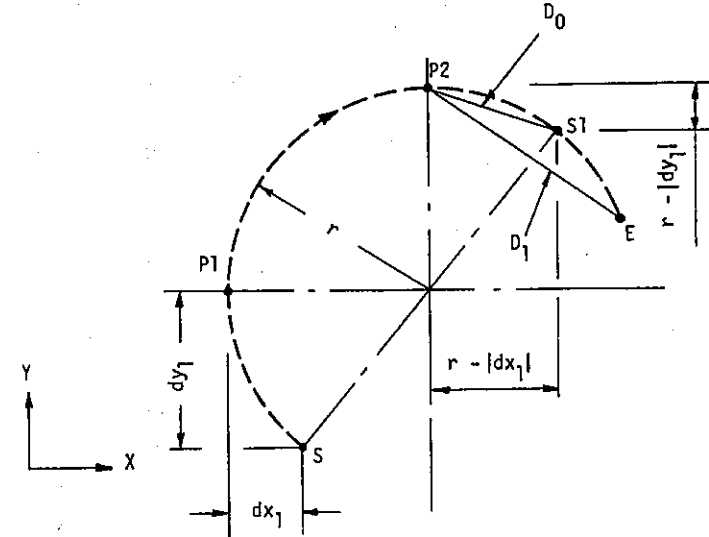
İÇİN İZİN VERİLEN İLERLEMELER, f (inç/dak.)

D'	0.005	0.001	0.0005	0.0001
5	406.6	181.8	128.6	57.5
1	181.8	81.3	57.5	25.7
0.5	128.6	57.5	40.7	18.2

I, J ve K merkez kaydırma kodları, önceden açıklandığı gibi yeniden hesaplanması gereken DATBUF(45) ile (47) tarafından tespit edilir; R kodu DATBUF(41) ile tespit edilir.

Dairesel hareketin yeri 180 dereceden küçük veya eşit bir açıya sahipse R kodun işareti pozitif olmalıdır, eğer açı 180 dereceden büyükse negatif olmalıdır (Bölüm 2.6.1'e bakınız). R'nin işaretini tespit etmek için yöntem şöyle olabilir: X-Y düzlemindeki dairesel hareketin üçüncü çeyreğinde olan başlama noktası, S, ve bitiş noktası, E, Şekil 14-9'da gösterildiği gibi olduğunu farz ediniz. R kodunun işareti, P2 ve S1 noktaları arasındaki (örneğin, D_0) ve P2 ve E arasındaki (örneğin, D_1) mesafeyi karşılaştırarak tespit edilebilir. Eğer $D_1 - D_0 < 0$ ise R kodu pozitif aksi takdirde, negatif olmalıdır. S, P1, P2 ve E noktalarının koordinatları DATBUF(4) ile (6), (42) ile (44), (48) ile (50), (54) ile (56) ve (1) ile (3) arasından tespit edilir. Böylece D_1 ve D_0 mesafeleri aşağıdaki gibi hesaplanabilir:

$$dx_1 = \text{DATBUF}(42)$$



Şekil 14-9 Dairesel enterpolasyon için R kodu işaretini tespiti.

$$dy_1 = \text{DATBUF}(43)$$

$$D_0 = [(r - dx_1)^2 + (r - dy_1)^2]^{\frac{1}{2}}$$

$$D_1 = \{[\text{DATBUF}(54)]^2 + [\text{DATBUF}(55)]^2\}^{\frac{1}{2}}$$

Dairesel hareketin üç noktası, üçüncü çeyrekte başkasına yerleştirilirse R kodunun işareti kolayca tespit edilebilir. Uç noktası birinci ve ikinci çeyreklerde ise pozitifdir, uç noktası dördüncü ve beşinci çeyreklerde ise negatiftir. Bu durumlarda hesaplama gerekli değildir. Sonuç olarak, R kodunun işaretini tespit etmek için program kısmı kolayca tasarlanabilir.

Halihazırdaki deyim kadar NC programın icrası için birikmiş zaman hesaplandıktan sonra işlem tamamlanır.

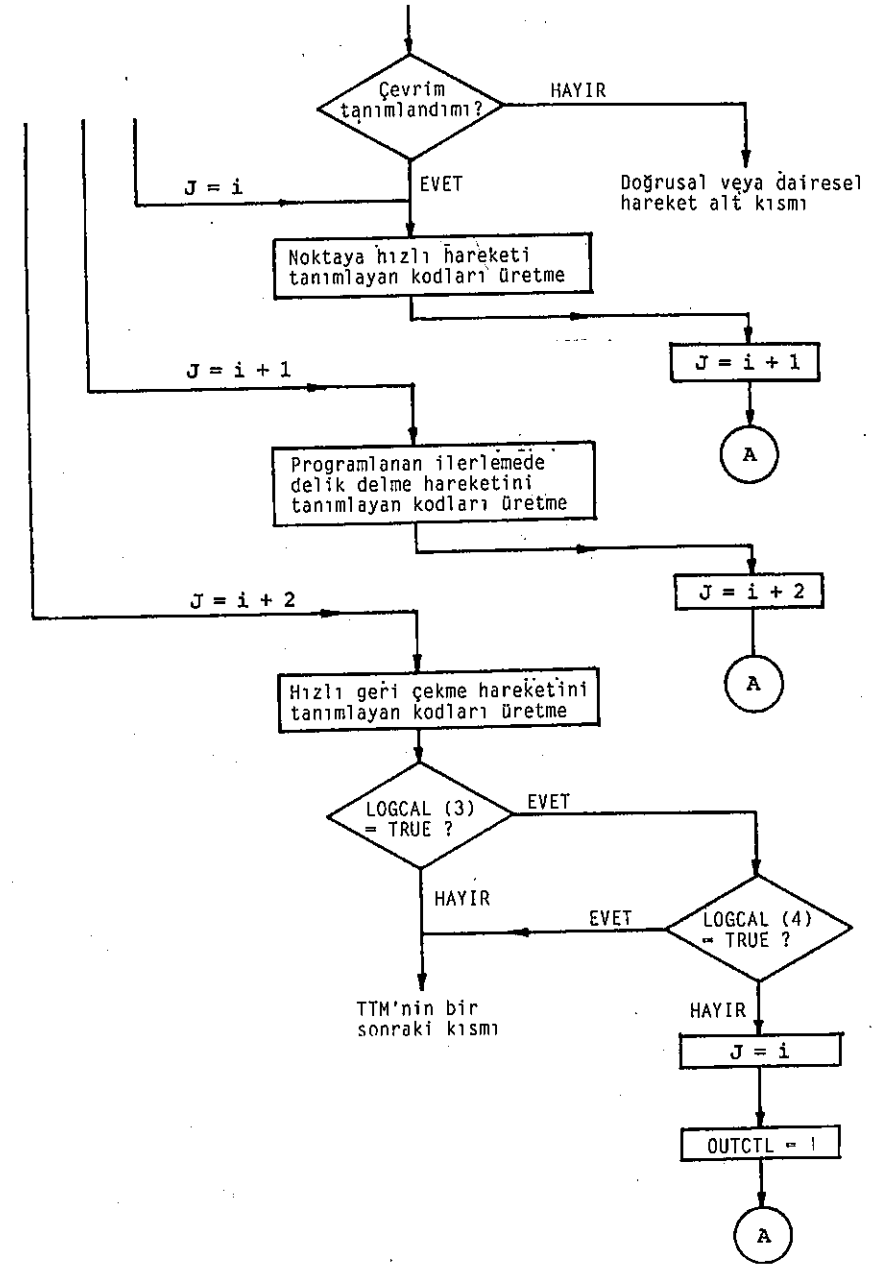
14.3.3.3 İşleme Çevrimi Alt Bölümü. Frezeleme çevrimi hariç, DAPP tarafından işlenebilen çevrim hareketi Z yönünde birkaç hareket kademelerinden ibarettir. MTM tarafından işlenmeden önce hareket çevrimi Bölüm 13.4.2.3'te açıklanan DATBUF elemanları tarafından tanımlanır. MTM'de bir çevrim hareketini işlemek için iki yol vardır: çevrim olmayan hareket kodlu çevrim hareketini tanımlama veya onu çevrim hareket kodları ile tanımlama.

İlk yaklaşım, çevrim hareketini G00 ve G01 NC kodlarını kullanarak tanımlamaktır. Böylece örneğin, basit bir delik delme çevrimi aşağıdakilere bölünebilir:

1. DATBUF(1) ile (3) tarafından tanımlanan delinecek noktaya hızlı hareket (G00)
2. Programlanan ilerlemede DATBUF(29) ile tanımlanan derinliğe delme hareketi (G01)
3. Uç noktasının DATBUF(30) ile tanımlandığı hızlı geri çekme hareketi (G00).

Böylece işlem, okunan her CLDATA grubu (veya DATBUF elemanları) için bir yerine üç NC hareket bloklarının türetilmesi gerektiği hariç, doğrusal enterpolasyon hareket alt bölümünde olanla çok benzerdir. Bu yöntem, işleme çevrim fonksiyonsuz NC tezgahları için kullanılabilir ve bu yordamın genel yapısı Şekil 1410'da gösterildiği gibi tasarlanabilir.

Bu yordam delinecek noktaya hızlı hareketi tanımlamak için NC kodlarını ayarlar ve J parametresi i + 1'e ayarlanır. Sonra kontrol, hızlı hareket bloğunu çıktı vermek için MTM'nin hazırlık bölümüne gönderilir. Programlanan ilerlemede delme hareketini tanımlayan kodları türetmeye başlayan DAPP'in Çıktı Bölümü kontrolü MTM'ye geri gönderir. Sonra kontrol J parametresi i + 2'ye ayarlandıktan sonra MTM'nin hazırlık bölümüne gönderilir. Bu kodlar çıktı olarak verildikten



Şekil 14-10 Çevrim hareket kodlarını işleyen alt bölümün genel yapısı. A: MTM'nin hazırlık bölümü.

sonra, geri çekme hareketini tanımlayan kodları türetmek için MTM tekrar çağrılır. Eğer çevrim basit bir delik delme veya delik büyültme çevrimi ise kontrol MTM'nin bir sonraki kısmına gider. Fakat eğer çevrim bir gagalı delik delme çevrimi

(BRKCHP veya DEEP) ise o zaman yukarıda açıklanan üç hareket kademesi önce başarısız olarak delme çevriminin delme çevrimini tanımlar ve diğer çevrimler bunu takip eder. Bu durumda, önce geri çekme hareketini tanımlayan kodlar çıkış olarak verilmeli ve çevrim işleme alt bölümü bir sonraki çevrim için NC kodlarını işlemek için tekrar çağrılmalıdır. Böylece yordam BRKCHP veya DEEP çevrimleri için bu alt bölümün başlangıcına işaret eden J'yi 1'e ve kontrolü bir sonraki çevrim verisinin grubunda okumak için yönlendiren OUTCTL'yi 1'e ayarlar. Bir karar deyimi, sonra halihazırdaki çevrimin başarısız delme çevriminde son olup olmadığını tespit etmek için konabilir. Eğer öyle ise, kontrol MTM'nin bir sonraki işleme bölümüne gitmelidir.

Frezeleme çevrimi de benzer şekilde işlenebilir.

Eğer son işlemci çevrim işleme fonksiyonlarına sahip olan bir NC'li tezgah için tasarlanırsa, MTM çevrim işleme kodlarını işleyebilmelidir; örneğin FANUC 6MB denetimi için G80 ile G89 arası kodlar. Bir G99 veya G98 kodu geri çekme hareketinin seviyesini tanımlamak için PREFUN deyimi ile tanımlanabilir; DATBUF(1) ile (3), (29) ve (30) arası değerler G81, G82, G84 veya G85 kodu ile tanımlanan hareket için X, Y, Z ve R kodlarını tanımlamak için kullanılabilir. Farklı delme derinliklerini temsil eden BRKCHP veya DEEP çevrim deyiminde tanımlanan bütün parametreleri bir NC bloğa taşımak mümkün değildir. Böylece son işlemci tasarımcısı, son işlemcinin Q kodu üretimi için takip etmesi gereken bir kural oluşturmalıdır. Örneğin, delme derinliği, q (Şekil 2-22'ye bakınız) şöyle seçilebilir:

$$q = 1.5 \cdot (\text{matkap çapı})$$

veya BRKCHP ya da DEEP çevrim deyiminde tanımlanan bir parametre ile tespit edilebilir.

14.3.4 Özel Fonksiyonları İşleme

İşleme özelliği ve kesici hareketi ile ilgili kodlara ilave olarak, bulunan konumda bekleme, bir program sonu, bir programın isteğe bağlı yarıda kesilmesi (veya isteğe bağlı durdurma), veya döner tablının dönüşü gibi özel fonksiyonları tanımlayan NC kodları da başarılı operasyon için gereklidir. Veriler DATBUF(18), (19), (22), (25) ve (26)'da saklanır. Verilerle ilgilenen alt bölümün fonksiyonu, bunları ilgili G veya M koduna veya eğer gerekirse (G04 kodundan sonra X kodu gibi) bazı diğer uygun kodlara dönüştürmektir.

Tablo 12-1'den görüleceği üzere, DATBUF(25)'teki değerler END, STOP ve DELAY gibi sıkça kullanılan APT son işlemci komutlarına atanırlar. APT'de PREFUN/ n deyimi ile tanımlanan özel bir hazırlık fonksiyonu DATBUF(25)'te

çevrilmelidir. Örneğin Gn . Diğer taraftan, APT'de ADXFUN/ n deyimi ile tanımlanan bir yardımcı fonksiyon DATBUF(25)'te bir pozitif skalardır, n ; bu ilgili M koduna çevrilmelidir, örneğin Mn .

Bu özel fonksiyonlarla ilgilenen bölümde, işleme özelliği kodlarını işleyen kısımdakine çok benzerdir. Fakat, bu mantıksal NC kod dizisi türetecek şekilde işleme yordamının tasarlanması gerektiğine dikkat edilmelidir. Örneğin, eğer bir M02 veya M03 program bitirme kodu türetilirse, yordam soğutucuyu ve iş mili durumunu inceleyebilmelidir ve eğer gerekiyorsa iş milini ve soğutucuyu durdurmak için M09 ve M05 kodlarını otomatik olarak türetebilmelidir. Bu yordam, denetimcinin sahip olmadığı tanımlanan özel fonksiyonlar için hata mesajları verebilmelidir.

Bu işleme yordamının genel yapısı 15. Bölümdeki örnekte görülebilir.

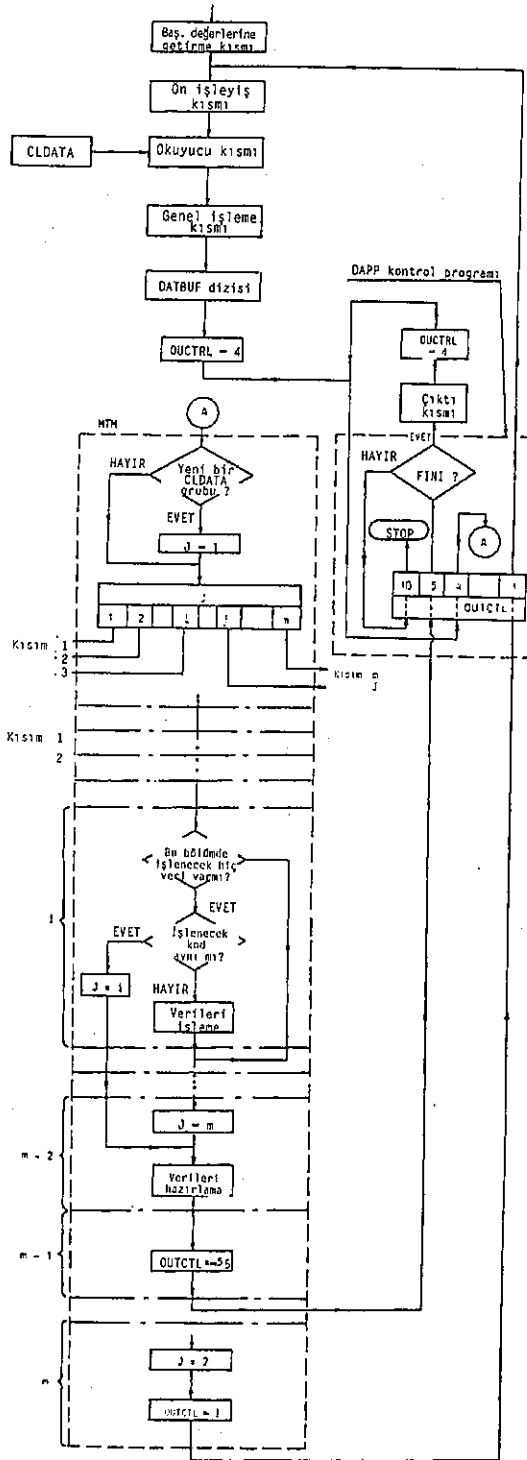
14.4 MTM'NİN HAZIRLAMA VE KONTROL-YÖNLENDİRME BÖLÜMLERİ

MTM'nin işleme bölümü tarafından işlenen veriler çıktılarını verilmeden önce hazırlama bölümüne gönderilir. Bu bölümün fonksiyonlarından birisi çıktısı verilecek bir kodu bir önceki bloktaki ilgili kodla karşılaştırmaktır. Eğer bu iki kod benzer ise, bu kodun bulunulan blokta çıktısının verilmesine gerek yoktur. Fakat, sadece kendilerinin tanımlandığı yerdeki blokta etkin olan NC kodları için (örneğin, G04, M11), yukarıda bahsedilen program bölümü atanmalıdır. Bir NC bloğun içeriğini son duruma getirdikten sonra, yordam çıktı kodlarının kaydını tutan diziyi yeniden düzenler. Denetim, MTM'den çıktıktan sonra denetimi yönlendiren denetim yönlendirme bölümüne gönderilir. Denetim yönlendirme bölümü iki alt kısma ayrılır. Önce alt kısım, çıktı ihtiyacına bağlı olarak OUTCTL'i 5, 6, 7 veya 11'e ayarlar. İkinci kısma sadece $J = k$ olduğunda girilir; bunun anlamı, bütün okunan verilerin işlendiği ve denetimin bütün işleme bölümlerine gittiğidir (Şekil 14-1'e bakınız). Sonuç olarak, denetim DAPP Ön İşlemler Bölümüne yönlendirilir ve sonra diğer bir grup CLDATA'yı okumak için Okuyucu Bölümüne yönlendirilir.

Denetim Ön İşlemler Bölümüne gitmeden önce, denetim yönlendirme bölümünün ikinci alt kısmı değişken tanımlama kısmında tekrar ayarlanmayan, fakat yeni bir CLDATA grubu işleneceği zaman yeniden ayarlanması gereken değişkenleri de tekrar ayarlamalıdır. Denetim yönlendirme bölümünün yapısı 15. Bölümde verilen örnekte görülebilir.

14.5 DAPP TABANLI SON İŞLEMCİDE İŞLEM AKIŞI

Şimdi MTM ve DAPP'a dayalı son işlemcide işlem akışını açıklayacağız. MTM'nin yapısı ve onun DAPP'ın çeşitli bölümleri ile ilişkileri Şekil 14-11'de gösterilmektedir. Bir grup CLDATA okunduktan, işlendikten ve DATBUF dizisi



Şekil 14-11 MTM ve DAPP'a dayalı son işlemcide işleme akışı. MTM'nin değişken tanımlama bölümünün gösterilmediğine dikkat ediniz.

sinde saklandıktan sonra MTM çağrılır. MTM'nin değişken tanımlama bölümü (gösterilmemekte) ve MTM'nin ilk değer verme bölümü icra edilir. Daha sonra DATBUF dizisindeki veriler MTM'nin farklı işleme bölümleri tarafından işlenir. Her bölümün başlangıcında, eğer DATBUF dizisinde bulunan bölüm tarafından işlenecek veri yoksa bir karar deyimi denetimi bir sonraki kısma dallandırır. Sonra eğer halihazırdaki kısım tarafından işlenecek kod bir NC blokta tekrarlanacaksa diğer karar değimi denetim hazırlama bölümüne (Şekil 14-11'deki $m - 2$ kısmına) dallandırılır. Eğer öyle ise, J parametresi ayarlanır ve denetim gerekli hazırlık işlemini yapan ve OUTCTL değişkenini 5, 6, 7 veya 11'e ayarlayan hazırlık bölümüne gider. Daha sonra denetim bir NC blok çıktısı vermek için Çıktı Bölümüne gönderilir. Çıktı Bölümü OUTCTL değişkenini 4'e ayarlar ve denetimi MTM'ye geri gönderir. Hesaplanan GOTO deyimi denetimi doğrudan önceden çıkmış olduğu işleme bölümüne aktarır. DATBUF dizisindeki bütün veriler işlene kadar, işlem bir bölümden diğer bir bölüme devam eder. Bu durumda, hazırlama kısmı başlangıçtan ($m - 2$ bölümünde $J = m$ deyimi) girilir. Veriler hazırlanıp çıktı olarak verildikten sonra MTM tekrar çağrılır ve denetim OUTCTL ve J değişkenini sırası ile 1 ve 2'ye ayarlayan son bölüme (m kısmına) gider. Denetim DAPP'ın Ön İşlemler Bölümüne geri döner ve CLDATA'nın yeni bir grubu çağrılır. Bu işleme çevrimi tüm APT programının CLDATA'sı işlene kadar tekrarlanır.

PROBLEMLER

- 14.1 Genellikle bir MTM'nin işleme bölümü, her biri NC tezgahının bir özel fonksiyonu ile ilgili verileri işleyen, uygun bir sırada düzenlenmiş bir kaç alt bölümden oluşur. Üç eksenli (döner tablasız) bir freze tezgahı için genellikle MTM'nin işleme bölümü kaç alt bölüme ayrılabilir? MTM'de bu alt bölümlerinden sırasını da açıklayınız.
- 14.2 2. Bölüm'de açıklanan Cadillac NC-100 tornası dört torna takımını tutan bir başlığa sahiptir. Kendini dört rakamın takip ettiğini bir T kodu kullanılan kesiciyi ve onun kaydırmasını açıklamak için gerekmektedir. Başlık sadece saatin aynı yönünde dönebilir. 90 derecelik saatin aynı yönü dönüş M11 kodu ile tanımlanmaktadır. Başlıktaki sırası ile dizilmiş kesiciler için tanıma numaraları saatin tersi yönünde (Şekil 6-11a) 1, 2, 3 ve 4'tür. Böylece bir veya birkaç M11 kodu, gerekli kesiciyi çalışma konumuna konumlamak için gerekebilir. Sadece LOADTL/n ($n = 1, 2, 3$ veya 4) deyiminin APT programında kesiciyi değiştirmek için kullanıldığını farz ediniz. Aşağıdaki varsayımlara dayalı olarak kesici değiştirme kodlarını (T ve M kodları) işlemek için ilgili işleme program bölümlerini (alt bölümleri) tasarlayın.
 - a. Bu program kısmı MTM'nin işleme programında ilk alt kısımdır.

- b. Bu program kısmı MTM'nin işleme programında ilk alt kısım değildir. Bu program kısımları için akış diyagramları hazırlayınız.
- 14.3 Bir NC'li tezgahın devir sayısının, dakikada devir sayısı olarak tanımlayan dört rakamın takip ettiği bir devir sayısı kodu, S ile tanımlandığını farz ediniz. Devir sayısını ve yön kodlarının işlenmesi için program bölümü (alt bölüm) tasarlayınız ve akış diyagramını çizin.
- 14.4 Bir NC'li tezgahın sadece iki soğutucu durumunun olduğunu, COOLNT/ON ve COOLNT/OFF farz ediniz. M08 ve M09 soğutucu kodlarını işlemek için program bölümünü tasarlayınız ve akış diyagramını çizin.
- 14.5 Modern SD'li tezgahlar için, bir doğrusal hareket genellikle parçalara ayrılmaz. Doğrusal ve noktadan-noktaya hareket kodlarını (G01, G01, X, Y ve Z kodları) işleyen bir program bölümünü tasarlayınız ve akış diyagramını çizin.
- 14.6 END, DELAY, OPSTOP ve STOP'u içine alan bir NC tezgahının özel fonksiyonlarını işleyen bir program bölümü tasarlayınız. Bu fonksiyonlar için ilgili NC kodları sırası ile M30, G04, M01 ve M00'dır.
- 14.7 Üç eksenli bir CNC freze tezgah denetleme sisteminin sadece aşağıda listelenen sınırlı fonksiyonlarına sahip olduğunu farz ediniz:

FONKSİYON	İLGİLİ KODLAR
Devir sayısı	dev./dak.'da devir sayısını temsil eden 4 rakamın kendini takip ettiği S kodu
Fener milini durdurma	M05
Fener mili açık (saatin ibresi yönünde)	M03
Fener mili açık (saatin ters yönünde)	M04
Soğutucu açık	M08
Soğutucu kapalı	M09
Üç eksenli (X, Y ve Z) doğrusal interpolasyon hareketi	G01X_Y_Z_
Üç eksenli (X, Y ve Z) aynı anda hızlı hareket	G00X_Y_Z_
Bekleme	G04P_
Program sonu	M02
Program durdurma	M00
İlerleme hızı	Son ikisinin ondalık rakam olduğu 5 rakamın kendini takip ettiği F kodu

Bu tezgahta otomatik takım değiştirici yoktur ve böylece T koduna gerek duyulmaz. (1) Kod formatlarının, tezgah ve denetimci parametrelerinin ve çıktı ihtiyaçlarının DAPP sorularına cevaplar tarafından uygun bir şekilde ayarlandığını ve (2) BLOCK DATA programının üretildiğini farz ediniz. Kod formatlarını tanımlayan OUTGRD elemanları aşağıdaki gibidir:

OUTGRD (i,j), j = 1 İLE 10 ARASI
İÇİNDEKİ İ DEĞERİ

OUTGRD (i,j), j = 1 İLE 10 ARASI İÇİNDEKİ İ DEĞERİ	KOD
1	G
2	X
3	Y
4	Z
5	F
6	S
7	P
8	M

Çıktı X, Y ve Z kodlarının mutlak koordinatlar olması gerektiğini varsayınız. İlk kesici pozisyonunun, x_1 ve y_1 'in MACHIN deyiminde DATBUF(34) ve (35) iki parametresi ile tanımlandığı ve z_1 'in 1 inç olduğu (x_1, y_1, z_1) konumu olduğunu kabul ediniz. Yukarıda listelenen tüm fonksiyonlarla ilgilenen bir tezgah için bir MTM tasarlayınız. Sonra tam bir akış diyagramı çizin.

- 14.8 Eğer bir grup CLDATA'dan farklı kodlar türetililebilirse, bir MTM çıktı NC bloklarının içeriğini nasıl tespit eder?
- 14.9 Eğer farklı grupların iki G kodu bir NC blokta izin verilirse MTM'nin tasarlanmasında hangi ölçü esas olarak alınmalıdır?

Bölüm 15

DAPP Tabanlı Dayalı Bir Son İşlemcinin Oluşturulması: Bir Örnek

11 ile 14. Bölümler arasında, CLFILE (CLDATA)'nın yapısını ve DAPP'ın çeşitli kısımlarının fonksiyonlarını açıkladık. DAPP'a dayalı son işlemcide kullanılan temel değişkenler, tezgaha bağlı değişkenlere ilk değer atama yöntemi ve takım tezgahı modülü (MTM) tasarlanmanın yöntemi de detaylı olarak açıklandı. Bundan dolayı okuyucu DAPP'a dayalı bir son işlemci tasarlamak için gerekli bilgiye sahiptir.

Bu bölümde, önce DAPP'a dayalı bir son işlemci tasarlamak için kullanılan yöntemleri tanıttıktan sonra bir SD'li tornayı, son işlemcinin tasarlanması ve uygulanmasını detaylı olarak açıklamak için bir örnek olarak kullanmaktayız.

15.1 DAPP TABANLI BİR SON İŞLEMCİNİN OLUŞTURULMASI İÇİN KULLANILAN İŞLEM SIRASI

DAPP'a dayalı bir son işlemcinin tasarlanması için işlem sırası aşağıdaki gibidir:

1. FORTRAN programında BLOCK DATA'yı türetmek için 12. Bölümde açıklanan yöntemi kullanınız. Örneğin, VM/CMS ortamında, bu türetilir ve dosya adı, tipi ve modu ile bir dosya olarak saklanır:

ANSWER FORTRAN A1

2. NC'li tezgah denetimi sisteminin özelliklerine dayalı olarak FORTRAN'da bir MTM yazınız. Örneğin, aynı ortamda, tasarlanan MTM şu dosya adı, tipi ve moduna sahip olabilir:

TTMD08 FORTRAN A1

3. Yukarıda listelenen iki dosyayı derlemek için FORTRAN-H derleyicisini kullanınız. Sonuçta ortaya çıkan iki dosya

ANSWER TEXT A1

ve

TTMD08 TEXT A1

dir.

4. Derlenen iki dosyayı DAPP ile bağla. VM/CMS ortamında aşağıdaki iş icra deyimleri (EXEC) LATH2PP adlı son işlemciyi türetmek için kullanılabilir:

&CONTROL ERROR

GLOBAL TXTLIB CMSLIB FORTLIBX

LOAD DAPPPP DLWSSR DAPCOM ANSWER MTMD08 (RESET DAPPPP

ORIGIN 30000)

GENMOD LATH2PP

PR LOAD MAP A5

&EXIT

Son işlemci yükleme modülü ve CLDATA'yı işlemek ve NC kodlarına dönüştürmek için kullanılabilen sonuç dosyası aşağıda verilmektedir:

LATH2PP MODULE A1

OS/MVS ortamında son işlemci türetmek için olan JCL (Job Control Language (İş Denetim Dili-İDD)) deyimleri Ek D'de verilmektedir. Eğer diğer isimler BLOCK DATA programı, MTM ve son işlemci için kullanılırsa, yukarıda gösterilen iş icra deyimlerindeki ilgili terimler uygun bir şekilde değiştirilmelidir.

5. Son işlemciyi kontrol ediniz. Son işlemcinin ilgilenebileceği bütün hareket ve işlemlerden ibaret olan işleme operasyonunu tanımlayan bir APT programı genellikle bir NC işlemci ve son işlemci tarafından işlenmelidir.

Son işlemci, işlenen sonuçlarla (NC kodları) tasarlanan işlem sırasını karşılaştırmakla ve çeşitli DATBUF elemanlarının değerleri ile işlenmiş NC kodunun karşılaştırılması ile doğrulanır.

15.2 CADILLAC NC-100 TORNA/FANUC 6T DENETİM SİSTEMİ İÇİN DAPP TABANLI BİR SON İŞLEMCİNİN TASARIMI VE UYARLANMASI

Şimdi 2. ve 12. Bölümlerde verilen Cadillac.NC-100 torna/FANUC 6T denetimci sistemini bir örnek olarak, bir son işlemcinin tasarlanması ve oluşturulmasına ek bilgi oluşturmak için kullanacağız.

15.2.1 Tezgah Kontrol Sisteminin Özellikleri

Bu tezgahın özellikleri aşağıdaki gibidir:

1. Z ve X eksenleri boyunca hareket
2. Artışlı veya mutlak koordinatlarda programlanan hareket
3. Programlanabilen takım seçimi
4. İki aralıkta ve iki yönde (saatin aynı ve ters yönü) programlanabilen devir sayısı
5. İnç'te programlama
6. Otomatik referans noktası dönüşü
7. Dakikada veya devirde inç cinsinden programlanabilen ilerleme hızı
8. Çeşitli fonksiyonlar

Bu denetimci hem ISO hemde EIA kodlarında girdi kabul eder ve aşağıdaki hazırlık ve yardımcı fonksiyonlarını sağlar:

M00	Programı durdurma
M01	İsteğe bağlı durdurma
M02	Program sonu
M03	Fener mili ileri doğru dönüşü (saatin tersi yönünde)
M04	Fener mili geri dönüşü (saatin aynı yönünde)
M05	Fener milini durdurma
M08	Soğutucu açık
M09	Soğutucu kapalı
M11	Saatin aynı yönünde döner başlığın 90 derece dönmesi
M30	Program sonu ve hafızayı geri sarma
M41	Devir sayısının düşük devirli aralığı

M42	Devir sayısının yüksek devirli aralığı
G00	Hızlı konumlama hareketi
G01	Doğrusal enterpolasyon hareketi
G02	Dairesel enterpolasyon hareketi (saatin aynı yönünde)
G03	Dairesel enterpolasyon hareketi (saatin tersi yönünde)
G04	Bekleme
G28	Referans noktasına geri dönme
G70-76	2. Bölümde açıklandığı gibi çoklu işleme çevrimleri
G90,92,94	Sırası ile tornalama, vida açma ve alın işleme çevrimleri
G98	Dakikada inç cinsinden programlanan ilerleme hızı
G99	Her devirde inç cinsinden programlanan ilerleme hızı

Bu denetimci Tablo 2-3'te verilen kelime adres formatlarında şu SD kodlarını kabul eder: N, G, X, Z, U, W, I, K, F, S, T, M, L, P, Q, D, A, fakat R yarıçap kodunu kabul etmez.

Aşağıda takdim edilen son işlemci, çevrim fonksiyon kodları (G90 ve G70'ler) hariç yukarıda listelenen bütün G kodları ile ilgilenebilir. Çünkü DAPP, tornalama ve çoklu tornalama çevrimlerinin işlenmesi için program kısımlarını sağlamaz. Onun için L, P, Q, D ve A kodları bu son işlemci tarafından ele alınmazlar.

Bu tezgahdaki döner başlık pnömatik tahriklidir. Dört konuma sahiptir ve sadece saatin aynı yönünde dönebilir. Döner başlıktaki dört takımın saatin tersi yönünde sıra ile 01, 02, 03 ve 04 diye numaralanmaktadır. Takım kaydırması, DAPP'ın geometrik kısmı tarafından kesici konumunun hesaplanması esnasında telafi edildiği için, her kesici için kesici kaydırma kodunun türetilmesine gerek yoktur. Böylece bu takım kaydırması son işlemci tarafından sifıra ayarlanır ve bu son işlemci tarafından üretilen kesici kodları T0100, T0200, T0300 ve T0400'dür.

APT'de kullanılan ölçü sisteminin inç olduğu kabul edilir. Ölçmelere göre, S devir sayısı kodu aşağıdaki denklemlere kullanılarak hesaplanabilir:

$$s = (w - 12)/2.38 \text{ (düşük devirli aralık: } 12 < w < 250 \text{ dev/dak)}$$

$$s = (w - 50)/19.5 \text{ (yüksek devirli aralık: } 250 \leq w < 200 \text{ dev/dak)}$$

burada w dakikada devir cinsinden gerçek devir sayısıdır.

Bu son işlemci ilerleme hızını devirde inç cinsinden çıktısını vermek için tasarlanmıştır çünkü, bu tornalama için bir ilerleme hızını tanımlamak maksadı ile yaygın olarak kullanılan birimdir. Fakat, DAPP genel işleme yordamı tarafından işlenen ilerleme hızı, yani DATBUF(10) daima dakikada inç cinsinden ifade edilir (Tablo 12-1'e bakınız). Devirde inç cinsinden ilerleme hızının çıktısı için, ilerleme hızı aşağıdaki deyim kullanılarak bir APT programında tanımlanmalıdır:

FEDRAT//

burada *f*, IPR küçük kelimesi olmaksızın devirde inç cinsinden ifade edilir. Ayrıca MACHIN deyimindeki üçüncü parametre 2 olarak tanımlanmalıdır; bu, devirde inç cinsinden tanımlanan ilerleme hızına göre işleme zamanını hesaplamak için özel bir program kısmını çalıştırır.

APT programında ve CLDATA'daki *x* ve *y* koordinatlarının sırası ile *z* ve *x* koordinatlarına dönüştürülmesi gerektiğinden bahsetmek önemlidir. NC programındaki *x* koordinatının çap değeri olması gerektiğine de dikkat edilmelidir (yani APT programındaki *x* koordinatının değerinin iki katı).

15.2.2 BLOCK DATA Programının ve MTM Hazırlanması

Bir son işlemcinin tasarımına başlamak için, DAPP soru listesine cevaplar hazırlanmalı ve sonra QUEST programını kullanarak BLOCK DATA programı türetilmelidir. Bu cevaplar ve BLOCK DATA programı sırası ile Şekil 12-3 ve 12-4'de listelenmiştir. Önerilen MTM altyordamı (TMTMD08) şekil 15-1'de verilmektedir ve onun çeşitli kısımları için akış diyagramları Şekil 15-2 ile 15-12 arasında verilmektedir.

Değişken Tanımlama Bölümü. Değişken tanımlama kısmında (Şekil 152) DAPCOM ve QUEST etiketli COMMON alanları tanımlanır; bunu bu MTM'ye özel değişkenler takip eder. Bu değişkenlerin anlamları aşağıdaki gibidir:

- RCD1 = Çıktı kodlarının kayıtlarını tutmak için 18 konumlu REAL*8 dizisi
- GQES = Değerinin sıfır olduğu G, X ve Z kodlarının çıktısı için kullanılan INTEGER*2 değişkeni
- INDX11 = DATBUF (73) ile (85) aralığında bir sonraki elde olan DATBUF elemanını gösteren INTEGER*2 değişkeni
- J = MTM'ye tekrar girildiğinde denetimin hangi kısma gönderileceğini gösteren INTEGER*2 değişkeni; ilk değer atama kısmının icrasını sağlamak için 1 ilk değerine getirilir.
- IQ = -10000 değerine getirilen INTEGER*2 değişkeni; OUTGRD (i,10) değerine dayalı olarak DATBUF(73) ile (85) arası elemanlardan bir çıktı üretmek için kullanılır.
- NOPRNT = Bir LOGICAL*1 değişkeni, bir çıktının (NOPRNT = .FALSE.) gerekip veya (NOPRNT = .TRUE.) gerekmediğini göstermek için kullanılır; .TRUE. olarak ilk değerine getirilir.

```

SUBROUTINE MIMD08
C*****
C
C MACHINE TOOL MODULE FOR NC-100 LATHE/FANUC 6T CONTROLLER SYSTEM
C*****
C
C SECTION 1 : VARIABLE DEFINITION
C*****
C DEFINITION OF COMMON DATA FIELDS
C*****
C IMPLICIT REAL*8 (A-H,O-Z)
COMMON /DAPCOM/
A ANULL ,CLEARS (3) ,CSURF ,CORREC (3) ,FEDIPR
B ,FEDMAX ,INNBUFF (252) ,ORGX ,ORGY ,ORGZ
C ,PNO (6) ,PTNOIP (11) ,RADI ,RDRUM (20) ,TABPOS
D ,THETA ,TLNTH (4) ,WORKR8 ,ZDRUM (20)
E ,AUXARR (6) ,CLKARG ,CLKHRS ,CLKMIN ,CLKSEC
F ,DHFED (10) ,DPHOL (10)
G ,CLKSW6 ,CONTRL ,DUMREC ,ERRCOD ,IDIREC
H ,IENDSW ,IFED1 ,IFED2 ,IFIPR ,IJSEQ
I ,ILIM ,IERCNT ,IERINC ,IERROR (20) ,INCBLK
J ,INSPUN ,IPLAB ,IPOINT ,IRECNO ,ISKIP
K ,ITAPL ,IWORDS ,JASGN ,JEND ,JLINE
L ,JP ,LNCIRC ,NBSTEP ,NOSTEP ,NOWRD
M ,NSST ,NSUBCL ,OUTCTL ,OUTROD ,PASS7
N ,PGCNT ,PRTARG ,PRTLIN (120) ,PUNBUF (144) ,PUNTIAP
O ,RESET5 ,SEQNM7 ,SEQINC ,ZPCTL1
REAL*8 INNBUFF
REAL*4 AUXARR ,CLKARG ,CLKHRS ,CLKMIN
A ,CLKSEC ,DHFED ,DPHOL
INTEGER INNBUFF ,CLKSW6 ,CONTRL ,DUMREC ,ERRCOD
A ,OUTCTL ,OUTROD ,PASS7 ,PGCNT ,PRTARG
B ,PUNTIAP ,RESET5 ,SEQNM7 ,SEQINC ,ZPCTL1
LOGICAL*1 PRTLIN ,PUNBUF
REAL*8 ORG (3)
EQUIVALENCE (ORG (1) ,ORGX)
REAL*4 DP ,CL ,DW
EQUIVALENCE (DP ,AUXARR (1)) , (CL ,AUXARR (2)) , (DW ,AUXARR (5))
LOGICAL*1 PTNO (88)
EQUIVALENCE (PTNO (1) ,PTNOIP (1))
COMMON /QUEST/
A CORTRS (3) ,DATBUF (85) ,HOME (3) ,LIMITS (11) ,MAXRAD
B ,PPCODE ,PPPARM ,TPLIMT ,ULINE0 (15) ,ULINE1 (15)
C ,ULINE2 (15) ,XCONVT ,YCONVT ,ZCONVT ,XTRA (2)
D ,TIME (8) ,ULINE3 (15) ,ULINE4 (15)
E ,CLDRMS ,CNTOUR ,DISKID ,DPDRMS ,DRUMZE
F ,IGOHOM ,IGOTAB ,IRPMOD ,IZHOR ,IDRLEN
G ,LEADSW ,NLINE ,OUTGRD (20,10) ,R10 ,STRISQ

```

Şekil 15-1 Cadillac NC-100 torna/FANUC 6T denetim sistemi için Takım Tezgahtı Modülü.

NOTL = Bir LOGICAL*1 değişkeni, kesici değişiminin (NOTL = .FALSE.) gerekip veya (NOTL = .TRUE.) gerekmediğini gösterir; .TRUE. olarak ilk değerine getirilir.


```

H      ,SIMOSW (4)  ,TABLGO      ,ZMOD (2)      ,Z10
I      ,CLKPRT    ,EOB          ,IEOB         ,IPART       ,IRSTO
J      ,LCHAR (2)  ,TABCOD      ,TAPPRT     ,TAPMET
REAL*8  LIMITS      ,MAXRAD
REAL*4  TIME        ,ULINE3      ,ULINE4
INTEGER*4 CLDRMS    ,CYTOUR     ,DISKID     ,DPDRMS
A      ,DRUMZE    ,OUTGRD    ,R10       ,STRISQ    ,SIMOSW
B      ,TABLGO    ,ZMOD       ,Z10
LOGICAL*1 CLKPRT    ,EOB          ,IEOB         ,IPART
A      ,IRSTO    ,LCHAR      ,TABCOD     ,TAPPRT     ,TAPMET
EQUIVALENCE (ZDEPTH,DATEBUF (29)) , (ZFINAL,DATEBUF (30))
REAL*8  CONVEN (3)
EQUIVALENCE (CONVEN (1),XCDIG,XCONVT) , (YCDIG,YCONVT) , (ZCDIG,ZCONVT)
INTEGER  ASSIGN (2)      ,CYCTYP (2)
EQUIVALENCE (ASSIGN (1),DATEBUF (20)) , (CYCTYP (1),DATEBUF (28))
LOGICAL*1 LOGCAL (6)     ,OUTSUX (40)
EQUIVALENCE (LOGCAL (1),DATEBUF (31)) , (OUTSUX (1),OUTGRD (1,1))
*****
C
C      CURRENT STATUS OF MACHINE TOOL IS DEFINED BY ARRAY RCD1
C
C      RCD1 (1) = G
C      RCD1 (2) = X
C      RCD1 (3) = Z
C      RCD1 (4) = I
C      RCD1 (5) = K
C      RCD1 (6) = F
C      RCD1 (7) = S
C      RCD1 (8) = T
C      RCD1 (9) = M
C      RCD1 (10) = U
C      RCD1 (11) = W
C      RCD1 (12) = RAPID FEEDRATE SWITCH
C      RCD1 (13) = SPINDLE DIRECTION
C      RCD1 (14) = SPINDLE RPM
C      RCD1 (15) = TOOL CODE
C      RCD1 (17) = COOLANT OUTPUT CODE
C      RCD1 (18) = SPINDLE SPEED RANGE
*****
C
C      VARIABLE SPECIFIC TO THIS MIM
C
*****
REAL *8 RCD1 (18)
REAL *4 LL1,SS,III
LOGICAL *1 NOTL, NOPRNT
INTEGER *2 INDX11,J,IQ,GQES

```

Şekil 15-1 (devamı var)

Gerekli DATEBUF konumlarının listeleme çıktısında çıktılarını verebilmesi maksadı ile DBFPRT etiketli ortak alan da bu kısımda tanımlanır.

İlk Değer Atama Bölümü. İlk değer atama kısmının fonksiyonu iki kademelidir (Şekil 15-3). İki NC deyimini NC çıktısında ilk iki deyim olarak türetmelidir:

```

INTEGER *4 G,X,Z,I,K,F,S,T,M,U,W
DATA NOTL/T/,IQ/-10000/,INDX11/73/,J/1/,NOPRNT/T/
*****
COMMON /DBFPRT/ N1,ISTART (5),LENGTH (5)
EQUIVALENCE (G,OUTGRD (2,10)) , (X,OUTGRD (3,10)) ,
1      (Z,OUTGRD (4,10)) , (I,OUTGRD (5,10)) ,
2      (K,OUTGRD (6,10)) , (F,OUTGRD (7,10)) ,
3      (S,OUTGRD (8,10)) , (T,OUTGRD (9,10)) ,
4      (M,OUTGRD (10,10)) , (U,OUTGRD (11,10)) ,
5      (W,OUTGRD (12,10))
*****
GQES=0
*****
50  GO TO (100,200,300,400,500,360,700,800,900,550,261,250,130,140,
1 375,330) , J
*****
C
C      ARGUMENT "J" DETERMINES THE SECTION TO BE USED
C
C      J      LABEL      SECTION
C      1      100      INITIALIZATION
C      2      200      COOLANT AND TOOL
C      3      300      ROTATION DIRECTION OF SPINDLE
C      4      400      LINEAR INTERPOLATION
C      5      500      CIRCULAR INTERPOLATION
C      6      360      ROTATION SPEED OF SPINDLE
C      7      700      CONTROL FUNCTION AND FEEDRATE
C      8      800      PREPARATION SECTION
C      9      900      CONTROL DIRECTING SECTION (2)
C      10     550      SUBSEQUENT QUADRANTS OF CIRCULAR
C      11     261      INTERPOLATION
C      12     250      TOOL CHANGE
C      13     130      TOOL CODE
C      14     140      GENERATING G50 CODE
C      15     375      SPECIFYING DATEBUF POSITIONS TO BE
C      16     330      PRINTED
C      17     375      GENERATING M03 OR M04 CODE
C      18     330      SET SPINDLE SPEED RANGE
*****
C
C      SECTION 2 : INITIALIZATION
C
*****
100 DO 110 J=1,18
110 RCD1 (J)=ANULL
DATEBUF (73)=28
DATEBUF (74)=0
DATEBUF (75)=0
OUTGRD (2,10)=73
OUTGRD (11,10)=74
OUTGRD (12,10)=75
OUTCTL=6
J=13
RETURN
130 DATEBUF (73)=50
IF (DATEBUF (27).NE.ANULL) GOTO 138
DATEBUF (27)=0.0D0
132 IF (DATEBUF (34).NE.ANULL) GOTO 135
DATEBUF (34)=0.0D0
135 OUTGRD (2,10)=73

```

Şekil 15-1 (devamı var)

```

OUTGRD(3,10)=27
OUTGRD(4,10)=34
OUTCYL=6
J=14
RETURN
138 DATBUF(27)=2*DATABUF(27)
GOTO 132
140 NI=4
ISTART(1)=1
ISTART(2)=24
ISTART(3)=36
ISTART(4)=72
LENGTH(1)=17
LENGTH(2)=2
LENGTH(3)=12
LENGTH(4)=9
DATABUF(72)=0.0
DATABUF(73)=ANULL
DATABUF(74)=ANULL
DATABUF(75)=ANULL
DATABUF(27)=ANULL
DATABUF(34)=ANULL
C*****
C
C SECTION 3 : COOLANT AND TOOL
C*****
200 IF (DATABUF(35).EQ.1.0D0.OR.DATABUF(35).EQ.2.0D0) CALL DLWDBF
IF (DATABUF(17).EQ.ANULL) GOTO 250
IF (DATABUF(17).EQ.RCD1(17)) GOTO 250
IF (DATABUF(17).LE.4.0D0) DATABUF(17)=4.0D0
DATABUF(17)=DATABUF(17)+4.0D0
M=17
RCD1(17)=DATABUF(17)
J=12
GO TO 801
C*****
250 IF (DATABUF(15).EQ.RCD1(15).OR.DATABUF(15).EQ.ANULL) GOTO 300
DO 252 I=1,4
IF (DATABUF(15).EQ.(1.0*L)) GO TO 260
252 CONTINUE
CALL DLWDR (6510)
GOTO 300
270 T=15
NOTL=.FALSE.
J=3
272 CLKARG=CLKARG+TIME(5)
GO TO 801
260 IF (RCD1(15).EQ.ANULL) GO TO 275
LLL=DATABUF(15)-RCD1(15)
LLL=IFIX(LLL)
IF (LLL.LT.0) LLL=4+LLL
LLLL=1
261 M=11*IQ
IF (LLLL.EQ.LLL) GO TO 265
J=11
LLLL=LLLL+1
GO TO 272
265 RCD1(15)=DATABUF(15)
DATABUF(15)=100*DATABUF(15)
GO TO 270

```

Şekil 15-1 (devamı var)

```

275 RCD1(15)=DATABUF(15)
DATABUF(15)=100*DATABUF(15)
T=15
NOTL=.FALSE.
C*****
C
C SECTION 4 : DIRECTION AND SPEED OF SPINDLE ROTATION
C*****
C
C SPEED AND RANGE
C*****
300 VV=DATABUF(14)
IF ((VV.EQ.RCD1(14)).OR.(VV.EQ.ANULL)) GO TO 360
IF (RCD1(18).EQ.ANULL) GOTO 330
IF ((DATABUF(13).NE.RCD1(13)).AND.(RCD1(13).NE.ANULL)) GOTO 350
IF ((VV.GE.250).AND.(DABS(RCD1(18)-42.0).GT.0.5)).OR.
1 ((VV.LT.250).AND.(DABS(RCD1(18)-41.0).GT.0.5))) GOTO 350
IF (VV.GE.250.0) GO TO 319
GOTO 315
330 IF (VV.GE.250.0) GO TO 318
M=41*IQ
315 SS=((VV-12)/2.38)*IQ
S=IFIX(SS)
RCD1(18)=41
GOTO 320
318 M=42*IQ
319 SS=((VV-50)/19.5)*IQ
S=IFIX(SS)
RCD1(18)=42
320 RCD1(14)=VV
GOTO 360
350 M=5*IQ
RCD1(13)=3
RCD1(14)=0
CLKARG=CLKARG+TIME(4)
J=16
GOTO 801
C*****
C
C DIRECTION
C*****
360 IF ((DATABUF(13).EQ.RCD1(13)).OR.(DATABUF(13).EQ.ANULL)) GOTO 400
IF (M.EQ.0) GOTO 370
J=6
GO TO 801
370 IF (RCD1(13).EQ.ANULL) GOTO 375
IF (DATABUF(13).NE.RCD1(13)) GOTO 375
RCD1(13)=DATABUF(13)
III=DATABUF(13)
III=IFIX(III)
GOTO (372,372,390),III
372 J=15
GOTO 392
375 IF (DATABUF(13).EQ.2) GOTO 385
IF (DATABUF(13).EQ.3) GOTO 388
380 M=(DATABUF(13)+3)*IQ
GOTO 389
385 M=(DATABUF(13)+1)*IQ
GOTO 389

```

Şekil 15-1 (devamı var)

```

388 M=(DATEBUF(13)+2)*IQ
389 RCD1(13)=DATEBUF(13)
      J=4
      GOTO 396
390 J=4
392 M=5*IQ
      RCD1(13)=3
      RCD1(14)=0
396 CLKARG=CLKARG+TIME(4)
      GO TO 801
C*****
C
C      SECTION 5 : LINEAR INTERPOLATION AND QUICK POSITIONING
C*****
400 IF ((DATEBUF(7).EQ.0.0D0).AND.(DATEBUF(8).EQ.0.0D0)) GO TO 700
450 IF ((DATEBUF(36).NE.0.0).AND.(DATEBUF(36).NE.ANULL)) GO TO 500
      IF (DATEBUF(12).EQ.0.0D0) GO TO 452
      GQES=1
      G=0
      X=2
      Z=1
      GO TO 480
452 IF (RCD1(13).EQ.3.0) GOTO 490
455 IF (G.EQ.0) GO TO 460
      J=4
      GO TO 801
460 G=IQ
      IF (DATEBUF(10).NE.RCD1(6)) F=10
470 X=2
      Z=1
      IF (DATEBUF(1).NE.0.AND.DATEBUF(2).NE.0) GO TO 472
      GQES=1
472 IF (DATEBUF(35).NE.2.0) GOTO 480
      IF (DATEBUF(14).EQ.0.0) GOTO 478
      TIME(1)=TIME(1)/DATEBUF(14)
      GOTO 480
478 CALL DLWDR (6508)
480 CLKARG=CLKARG+TIME(1)
      J=7
      GO TO 801
490 CALL DLWDR (6501)
      DATEBUF(24)=1.0
      GO TO 455
C*****
C
C      SECTION 6 : CIRCULAR INTERPOLATION
C*****
500 IF (DATEBUF(12).NE.1.0) GO TO 510
      CALL DLWDR (6507)
      DATEBUF(24)=1.0
510 F=10
      IDTB=42
      NUMQD=0
      IF (G.EQ.0) GO TO 520
      J=5
      GO TO 801
520 G=2*IQ
      IF (DATEBUF(37).GT.1.0) G=3*IQ
550 U=IDTB+1
      W=IDTB

```

Şekil 15-1 (devamı var)

```

I=IDTB+4
K=IDTB+3
NUMQD=NUMQD+1
GOTO (553,556,558),NUMQD
553 DATEBUF(I)=DATEBUF(39)-DATEBUF(5)
      DATEBUF(K)=DATEBUF(38)-DATEBUF(4)
      GOTO 559
556 DATEBUF(I)=DATEBUF(39)-(DATEBUF(5)+DATEBUF(43)/2)
      DATEBUF(K)=DATEBUF(38)-(DATEBUF(4)+DATEBUF(42))
      GOTO 559
558 DATEBUF(I)=DATEBUF(39)-(DATEBUF(5)+DATEBUF(43)/2+DATEBUF(49)/2)
      DATEBUF(K)=DATEBUF(38)-(DATEBUF(4)+DATEBUF(42)+DATEBUF(48))
559 IF (DATEBUF(IDTB+6).EQ.ANULL.OR.NUMQD.EQ.3) GO TO 560
      J=10
      IDTB=IDTB+6
      GO TO 801
560 IF (DATEBUF(35).NE.2.0) GOTO 570
      IF (DATEBUF(14).EQ.0.0) GOTO 562
      TIME(1)=TIME(1)/DATEBUF(14)
      GOTO 570
562 CALL DLWDR (6508)
570 CLKARG=CLKARG+TIME(1)
      J=7
      GO TO 801
C*****
C
C      SECTION 7 : SPECIAL CONTROL FUNCTIONS AND FEEDRATE
C
C      THE FUNCTIONS HANDLED ARE:
C
C      FUNCTION                                DATEBUF(25)
C      -----                                -
C      END                                        9001
C      OPSTOP                                    9002
C      STOP                                       9003
C      DELAY                                     9004
C      PREFUN                                    -1 TO -99
C      AUXFUN                                    0 TO 99
C*****
700 IF (DATEBUF(25).EQ.ANULL) GO TO 790
      J=7
      IF (M.NE.0) GO TO 801
      IF (DATEBUF(25).GE.0.0) GO TO 710
      DATEBUF(25)=-DATEBUF(25)
      G=25
      GO TO 800
710 JJ1=DATEBUF(25)-9000.0
      IF (JJ1.GE.0.0) GO TO 720
      M=25
      GO TO 800
720 IF (JJ1.LE.1.OR.JJ1.GT.4) GO TO 770
      GO TO (730,740,750,760),JJ1
730 IF (RCD1(17).NE.8.0) GO TO 735
      M=9*IQ
      RCD1(17)=9.0
      GO TO 801
735 IF (RCD1(13).EQ.3.0) GO TO 738
      M=5*IQ
      RCD1(13)=3.0
      GO TO 801

```

Şekil 15-1 (devamı var)

```

738 M=2*IQ
GO TO 755
740 M=IQ
GO TO 751
750 M=72
751 RCD1(12)=0.0
RCD1(13)=3.0
RCD1(17)=9.0
755 CLKARG=CLKARG+TIME(4)
GO TO 800
760 G=4
U=19
GO TO 800
770 CALL DLWDER (6503)
DATEBUF(24)=0
790 IF (DATEBUF(10).NE.RCD1(6)) GOTO 800
IF (DATEBUF(10).EQ.0) GOTO 800
F=10
*****
C
C SECTION 8 : PREPARATION SECTION — PREPARING DATA,
C UPDATING OUTGRD & RCD1 ARRAY
C
*****
800 J=9
801 DO 840 I=2,12
IF (L.EQ.2.AND.G.EQ.4) GOTO 832
IF ((L.EQ.2).AND.(GQES.EQ.1).AND.(G.EQ.0)) GO TO 830
IF ((L.EQ.3).AND.(GQES.EQ.1).AND.(DATEBUF(2).EQ.0)) GO TO 830
IF ((L.EQ.4).AND.(GQES.EQ.1).AND.(DATEBUF(1).EQ.0)) GOTO 830
IF (OUTGRD(L,10)) 802,840,810
802 DATEBUF(INDX11)=(OUTGRD(L,10))/IQ
OUTGRD(L,10)=INDX11
INDX11=INDX11+1
810 IF (L.EQ.5.OR.L.EQ.6.OR.L.EQ.11.OR.L.EQ.12) GOTO 812
IF ((L.EQ.10).AND.(DATEBUF(OUTGRD(L,10)).EQ.11)) GOTO 816
814 IF (DABS(DATEBUF(OUTGRD(L,10))-RCD1(L-1)).LT.0.00001) GOTO 818
816 RCD1(L-1)=DATEBUF(OUTGRD(L,10))
IF (L.EQ.3.OR.L.EQ.11) GOTO 817
819 NOPRNT=.FALSE.
GOTO 840
817 IF (L.EQ.11.AND.RCD1(1).EQ.4) GOTO 819
DATEBUF(OUTGRD(L,10))=2*RCD1(L-1)
GOTO 819
812 RCD1(L-1)=0
GOTO 814
818 OUTGRD(L,10)=0
GOTO 840
830 DATEBUF(INDX11)=0
GOTO 835
832 DATEBUF(INDX11)=4.0
835 OUTGRD(L,10)=INDX11
INDX11=INDX11+1
GOTO 816
840 CONTINUE
*****
C
C SECTION 9 : CONTROL-DIRECTING SECTION (1)
C
*****
INDX11=73

```

Şekil 15-1 (devamı var)

```

IF (NOPRNT) GO TO 50
OUTCTL=5
NOPRNT=.TRUE.
GQES=0
RETURN
*****
C
C CONTROL DIRECTING SECTION (2)
C
*****
900 IF (DATEBUF(15).LT.10) GOTO 901
DATEBUF(15)=DATEBUF(15)/100
901 IF (DATEBUF(24).LT.0.5) GO TO 905
J=9
DATEBUF(24)=0.0
OUTCTL=12
RETURN
905 J=2
OUTCTL=1
DO 920 I=73,85
920 DATEBUF(I)=ANULL
RETURN
END

```

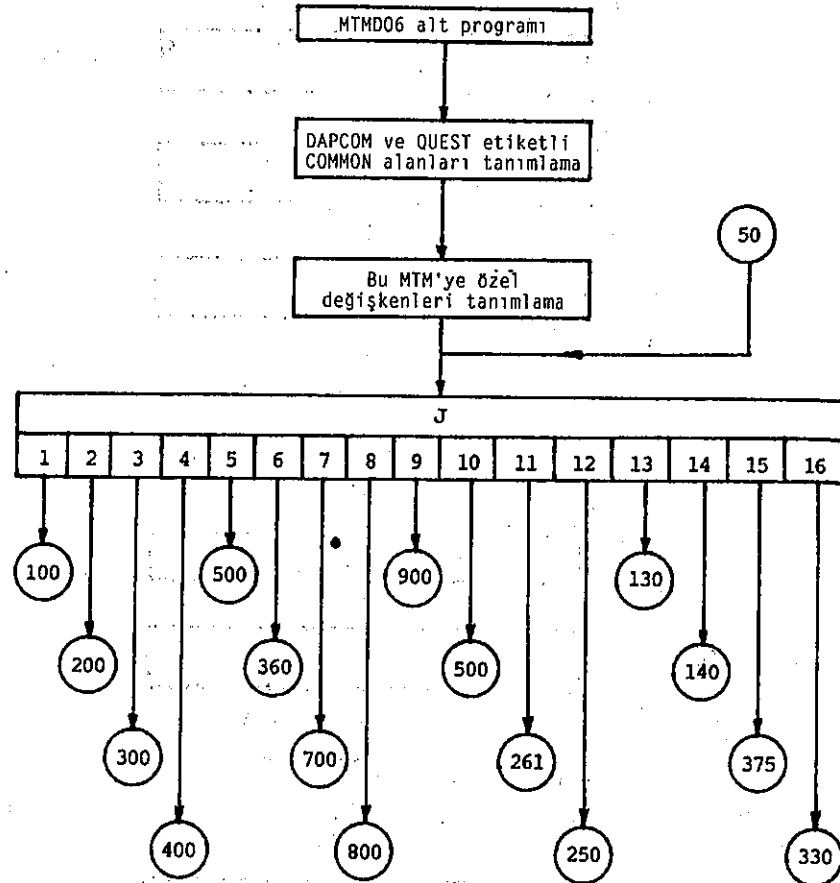
Şekil 15-1 (devamı var)

G28U0W0*
G50X_Y_*

İkinci deyimdeki X ve Z kodları sırası ile CUTCOM deyimindeki parametreden ve MACHIN deyimindeki ikinci parametreden alınır. Onun için, bir APT parça işleme programının programlanmasında bir ilk CUTCOM deyimini ve MACHIN deyimini tezgah koordinat sisteminin orijini tanımlamak için kullanılmalıdır. İlk değer atama kısmı RCD1 dizisinin ilk değerlerini ve listeleme dosyasında yazılacak DATEBUF elemanları da ayarlar. Son olarak, bu kısımda kullanılan DATEBUF elemanları da ANULL sembolü ile temsil edilen "girdi yok" durumuna yeniden ayarlanır.

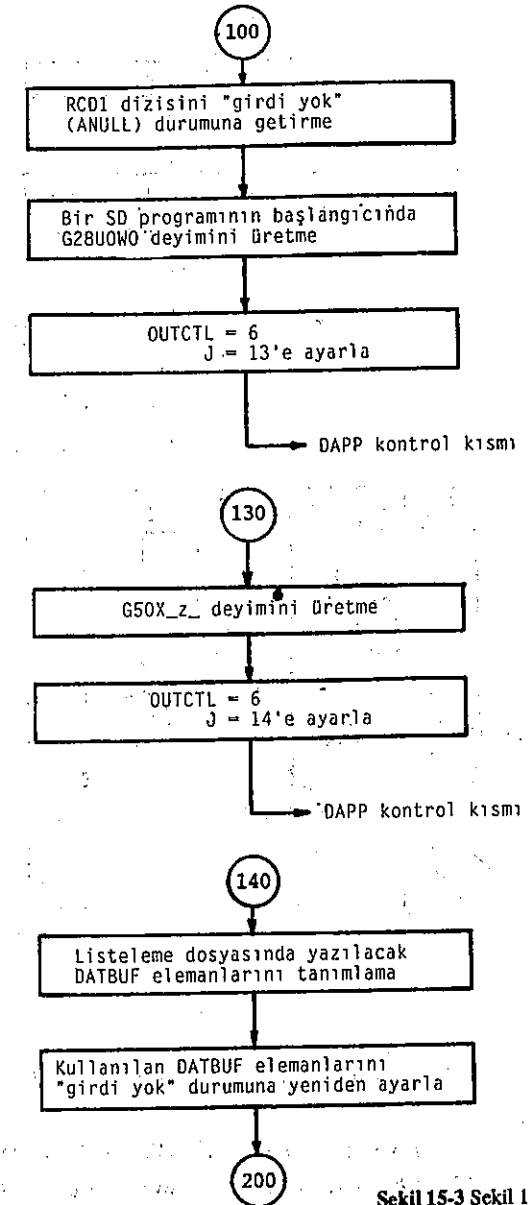
Soğutucu ve Takım Bölümleri. Soğutucu kısmı (Şekil 15-4) soğutucu verilerini işler. CLDATA kayıtları grubunun işlenmesi esnasında her MTM çağrıldığında ilk işleme kısmı girişleri denetler. Eğer MACHIN deyimindeki üçüncü parametre 1 veya 2 olarak tanımlanırsa, gerekli DATEBUF konumlarını yazmak için DATEBUF dizi yazma yordamı DLWDBF çağrılır. Soğutucu gerekirse, M08 kodu ayarlanır ve çıktısı verilir; sonra denetim MTM'ye tekrar girerek kesici kısmına gider. Aksi takdirde, denetim doğrudan kesici kısmına gider.

Kesici kısmının fonksiyonu (Şekil 15-5) gerekli kesici kodunu üretmek (T0100, T0200, T0300 veya T0400) ve döner başlığa saatin aynı yönünde 90 derece



Şekil 15-2 Şekil 15-1'deki MTM'nin değişken tanımlama bölümü için akış diyagramı.

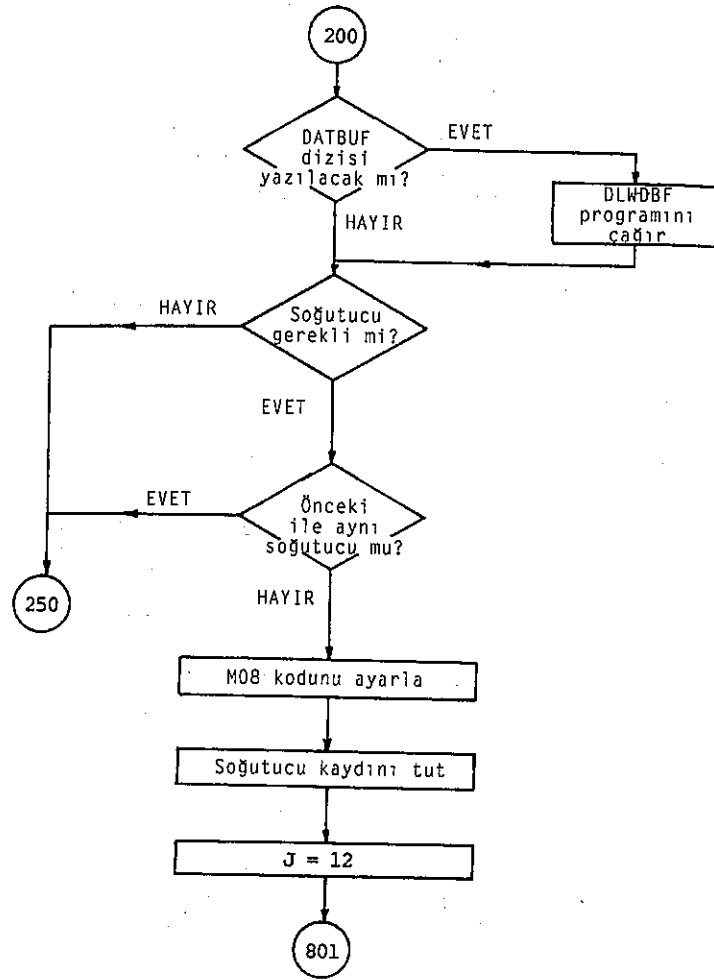
dönmesi için emir veren döner başlık dönme kodu (M11) üretmektir. Kesici değişimi LOADTL/n deyimi ile tanımlanır; burada n döner başlıkta bir sonraki konumdaki kesici olmayabilen, istenen kesicinin numarasıdır. Bir APT programında tanımlanan ilk kesicinin daima kesici No. 1 olduğu kabul edilir. Program gerekli M11 kodlarının sayısını hesaplar ve sonra istenen kesici kodunu ve M11 kodlarını üretir. Örneğin, halihazırdaki kesici No. 2 ise ve LOADTL/1 deyimi tanımlanmışsa (yani, No. 1 kesici istenmekte) kesici işleme yordamı üç NC bloğu üretir:



Şekil 15-3 Şekil 15-1'de listelenen MTM'nin ilk değer atama bölümü için akış diyagramı.

M11*
M11*
M11T0100*

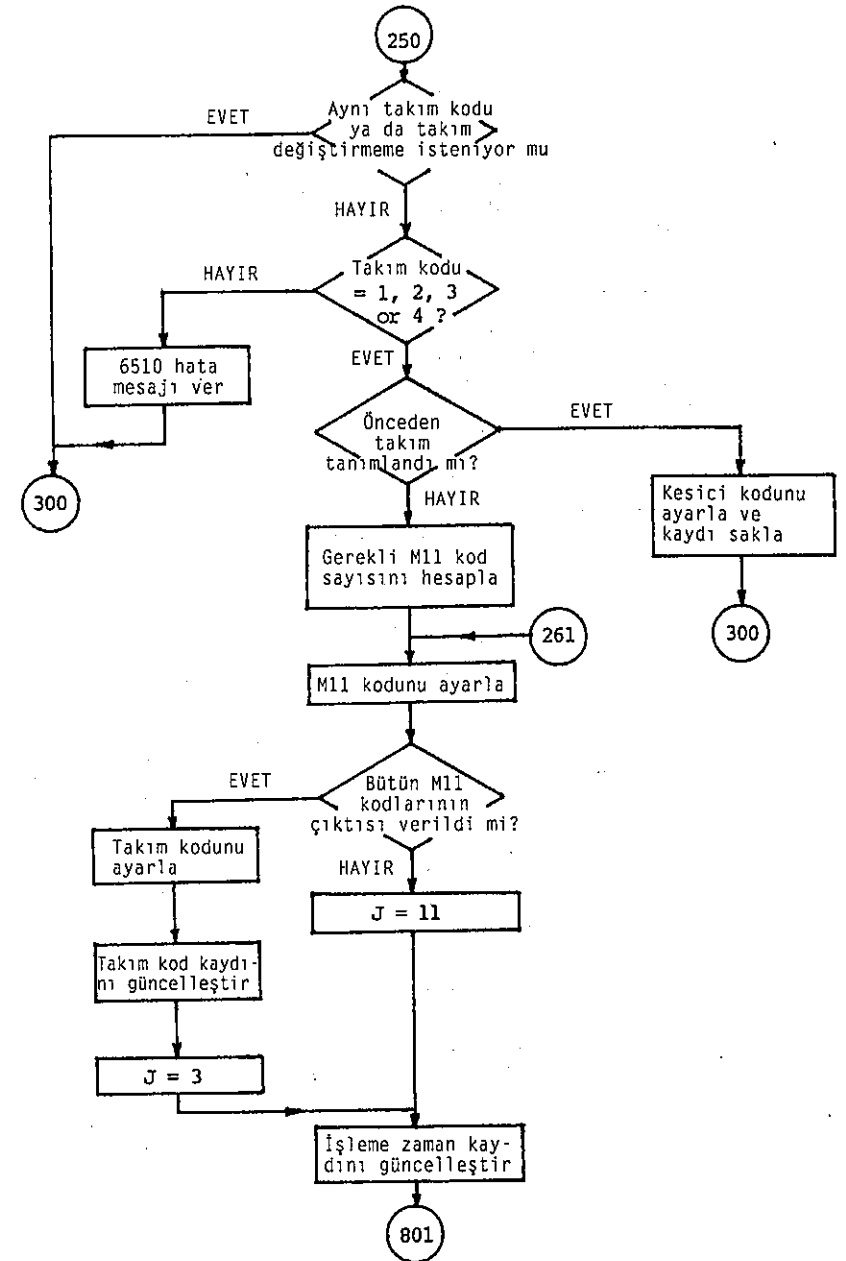
Sonuç olarak, NC tezgahı döner başlığı 270 derece döndürür ve kesici No. 1'i konumuna yerleştirir.



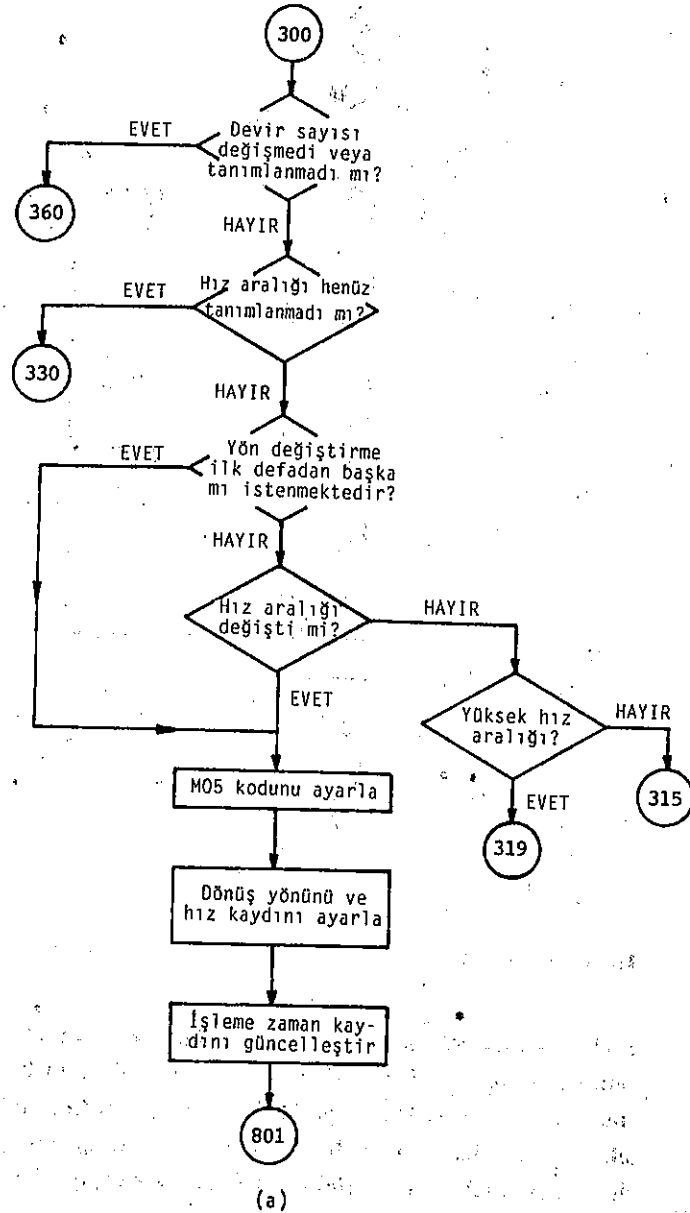
Şekil 15-4 Şekil 15-1'de listelenen MTM'nin soğutucu bölümü için akış diyagramı. 250: Kesici bölümündeki ilk deyim (Şekil 15-5). 801: Hazırlık bölümündeki ikinci deyim (Şekil 15-11).

LOADTL deyiminde tanımlanan kesici numarasının 1, 2, 3 veya 4 olduğu varsayılmaktadır; İlgili çıktı kodları T0100, T0200, T0300 veya T0400'dür. Bu DATBUF(15) değerinin çıktısı alınmadan önce 100 ile çarpılmış demektir. Onun için okuma CLDATA grubunun tamamının çıktısı alındıktan sonra DATBUF(15) denetim yönlendirme bölümünde orjinal değerine yeniden ayarlanmalıdır.

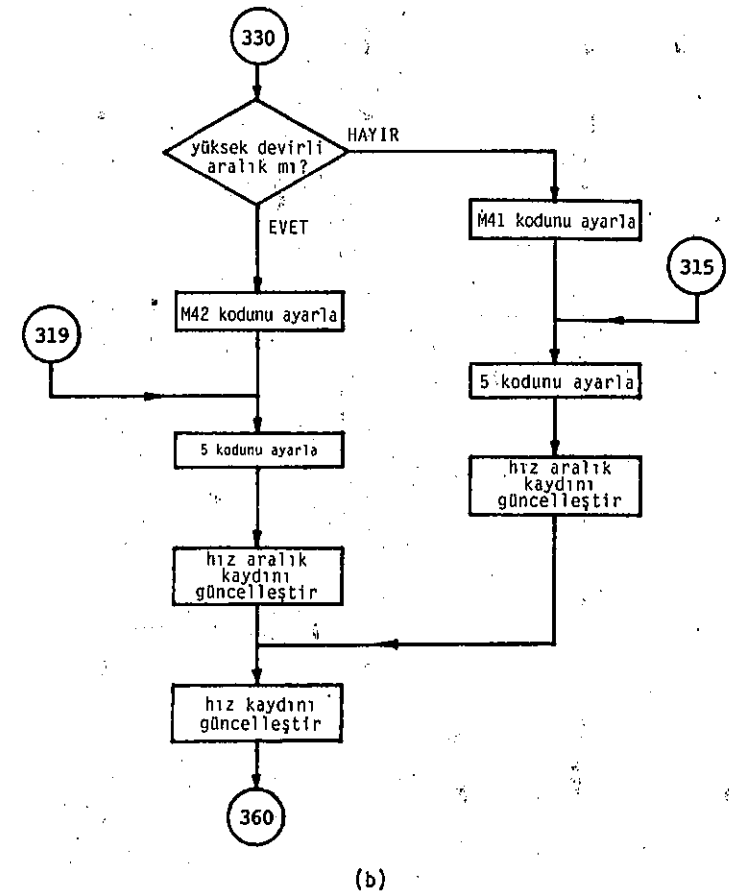
Bu bölümün sonunda, işleme zamanı kendine kesici değiştirme zamanı eklenerek güncelleştirilir.



Şekil 15-5 Şekil 15-1'de listelenen MTM'nin kesici bölümü için akış diyagramı.



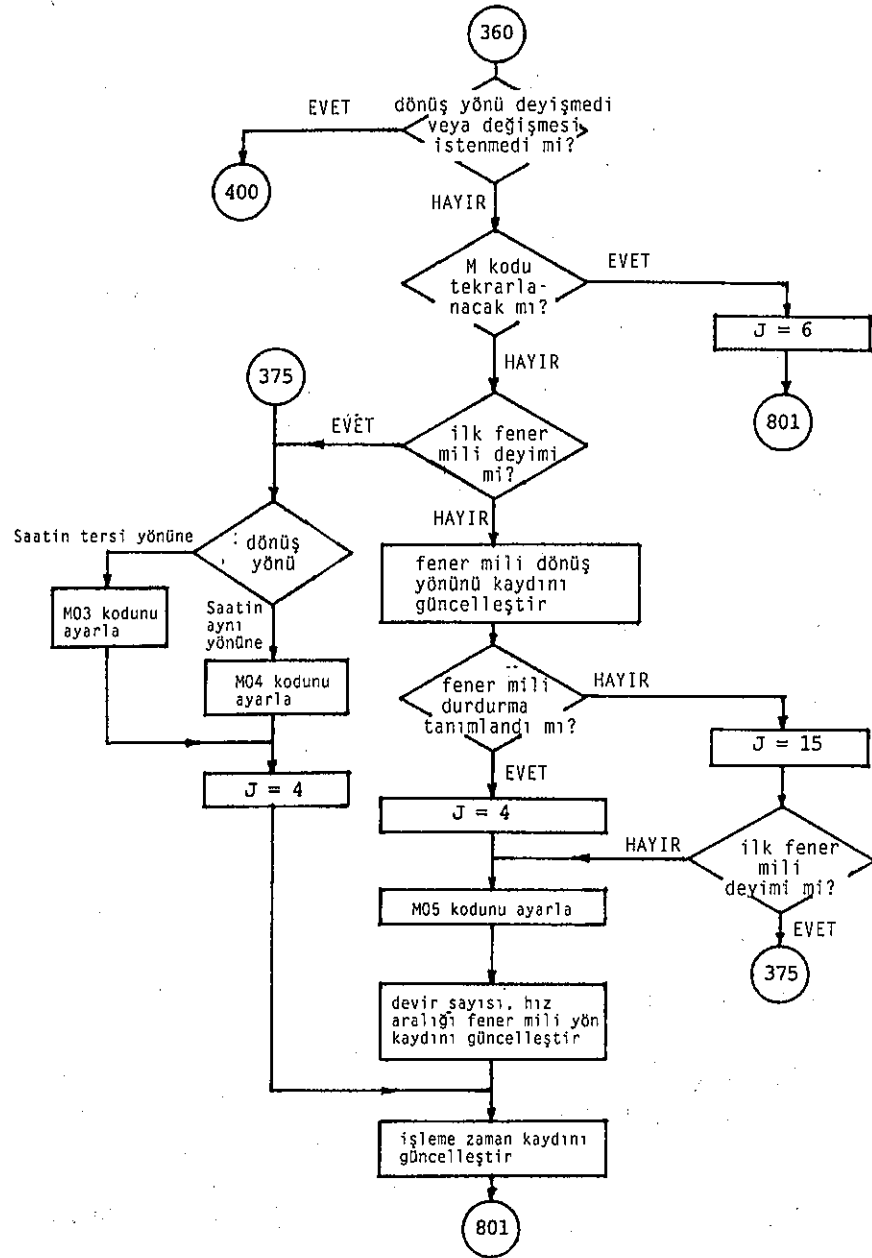
Şekil 15-6(a) Şekil 15-1'de listelenen MTM'nin iş mili dönme bölümü için akış diyagramı.



Şekil 15-6(b) Şekil 15-1'de listelenen MTM'nin devir sayısı ve aralığı bölümü için akış diyagramı.

İş Mili Dönüş Bölümü. Bu (Şekil 15-6[a ve b] ve 15-7) MTM'nin dördüncü bölümüdür; iş mili dönüş yönü ve hızı ile ilgili olan kodları işler. Bu kısımda türetilen NC kodları M41 veya M42 hız aralığı kodları ve M03, M04 veya M05 iş mili açık veya kapalı kodudur. Program APT'de tanımlanan bir SPINDL deyimini için gerekli kodları üretmelidir. Aşağıdaki durumlar gözönüne alınmalıdır:

1. Eğer SPINDL deyimini bir program içinde ilk defa tanımlanırsa, program bir M41 veya M42 hız aralığı kodu, bir S hız kodu ve bir M03 veya M04 iş mili yön kodu üretmelidir.



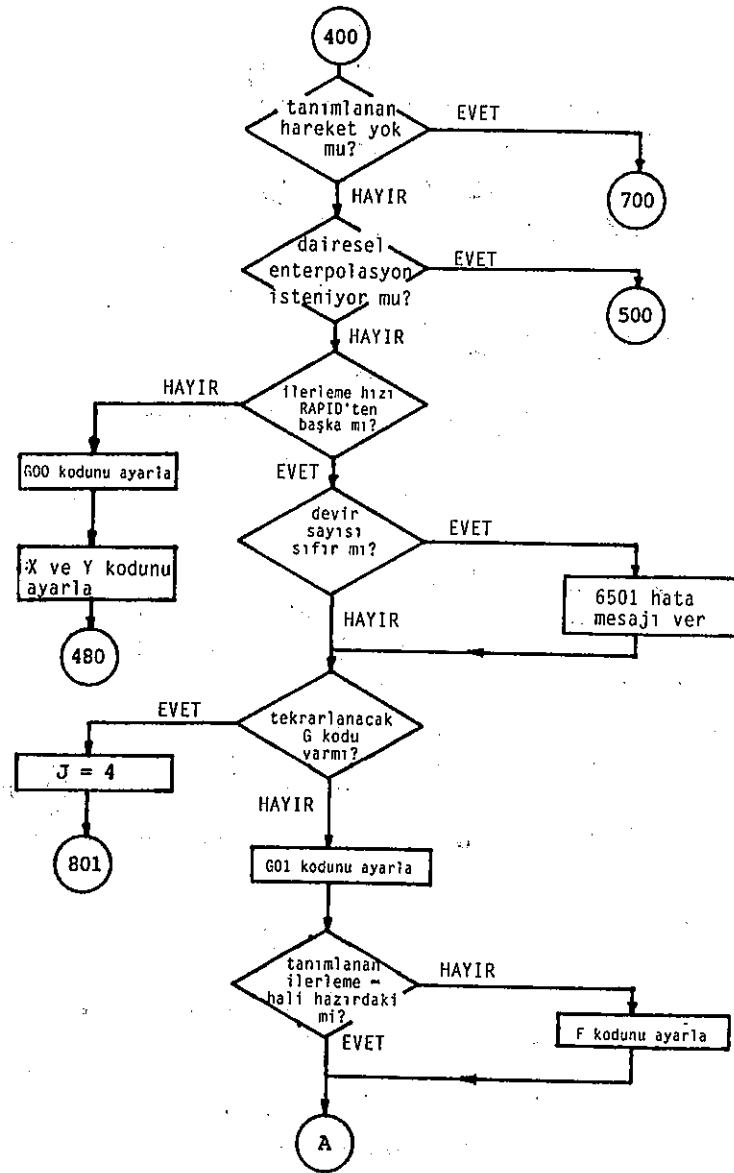
Şekil 15-7 Şekil 15-1'de listelenen MTM'nin iş mili dönüş yönü bölümü için akış diyagramı.

2. Eğer SPINDL deyimi bir programda ilk tanımlandıktan farklı ise program aşağıdakileri üretmelidir:
 - a. Eğer hem hız aralığı hem de hız değiştirildi ise sırası ile bir M05 kodu, bir M41 veya M42 kodu, bir M03 veya M04 kodu ve bir S kodu. (Bu tezgah önce fener milini durdurmalı, hızı değiştirmeli, hızı ayarlamalı ve sonra fener milini çalıştırmalıdır.
 - b. Eğer sadece hız aralığı değiştirildi ise, sırası ile bir M05 kodu, bir M41 veya M42 kodu ve bir M03 veya M04 kodu
 - c. Eğer sadece fener mili yönü değiştirildi ise, bir M05 kodu ve bir M03 veya M04 kodu
 - d. Eğer hız aralığı ve dönüş yönü değiştirilmedi ise sadece S hız kodu
 - e. Eğer SPINDL/OFF deyimi tanımlanırsa M05 kodu

Son olarak işleme zaman kaydı güncelleştirilir.

Doğrusal Enterpolasyon Hareket Bölümü. Doğrusal enterpolasyon hareket bölümü (Şekil 15-8) hızlı ilerlemede noktadan-noktaya hareketi ve doğrusal enterpolasyon hareketini işler. Bu kısımda türetilen kodlar G00 veya G01 kodu ile X, Z ve F kodlarıdır. Eğer tanımlanan ilerleme RAPID (yani hızlı) ise G00 kodu; aksi takdirde G01 kodu türetilir. G01 veya G00 kodunu tanımlayan DATBUF elemanı yoktur. Böylece OUTGRD (2,10) olan G değişkeni önce sıfıra veya IQ'ya (= -10000) ayarlanır; daha sonra hazırlık kısmında ayrılmış elemanlardan birine, DATBUF(73) ile (85) arası, aktarılır ve saklanır. Eğer doğrusal enterpolasyon hareketi tanımlanırsa ve devir sayısı sıfır ise 6501 hata mesajı türetilir. İlerleme hızı devirde inç olarak tanımlandığında eğer devir sayısı sıfır ise program 6508 hata mesajı verir. DAPP'ta hareket zamanını hesaplayan program dakikada inç cinsinden ilerleme hızı için tasarlandığından, ilerleme hızı devirde inç cinsinden tanımlandığında hareket zamanı yeniden hesaplanmalıdır. MACHIN deyiminin üçüncü parametresi, DATBUF(35), ilerleme hızının devirde inç cinsinden tanımlandığını MTM'ye veya son işlemciye bildirmek için 2'ye ayarlanmalıdır. Bu kısımdan çıkmadan önce, işleme zaman kaydı uygun bir şekilde güncelleştirilir.

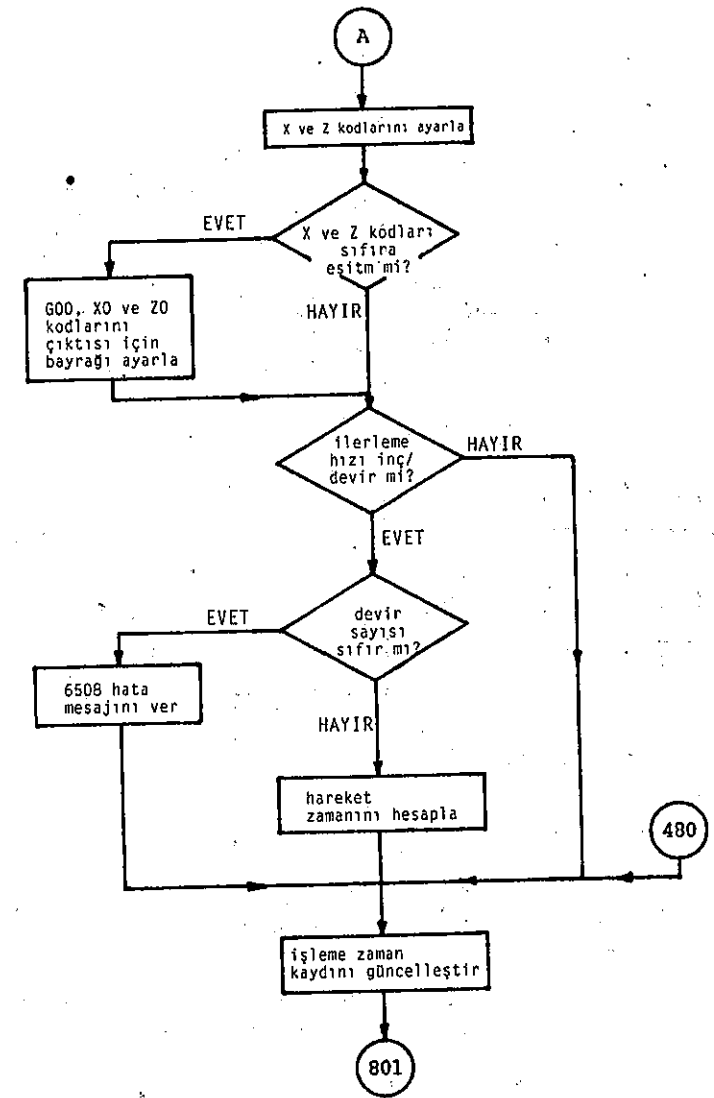
Dairesel Enterpolasyon Hareket Bölümü. Dairesel enterpolasyon hareket bölümü (Şekil 15-9) G02 veya G03 kodlarını daireysel hareket yönü için, U ve W kodlarını varış noktası için, I ve K kodlarını çemberin merkezi için ve F ilerleme kodunu işler. FANUC 6T denetimci için, daireysel bir hareket parçalanmadan tanımlanabilir. Fakat, bu programda daireysel hareket, daireysel hareketi farklı parçalarda yani DATBUF(42) ile (59) arası tanımlayan DATBUF elemanlarının değerini incelemek için farklı çeyreklerde farklı parçalara bölünür. Tornalamada daireysel hareket üç çeyreğe dağıtılamayacağı için en çok parça sayısı üçe ayarlanır. Çeşitli parçalar için uç koordinatları tanımlamak maksadı ile kullanılan DATBUF elemanları DATBUF(42), (43), (48), (49), (54) ve (55)'tir. Kullanıcı, parçalara ayrılmamış



Şekil 15-8 Şekil 15-1'de listelenen MTM'nin doğrusal interpolasyon hareket bölümü için akış diyagramı.

dairesel harekete sahip olmak isterse, sadece Z ve X kodlarını üretmek için sırası ile DATBUF(1) ve (2) tarafından tanımlanan uç noktasını kullanabilir.

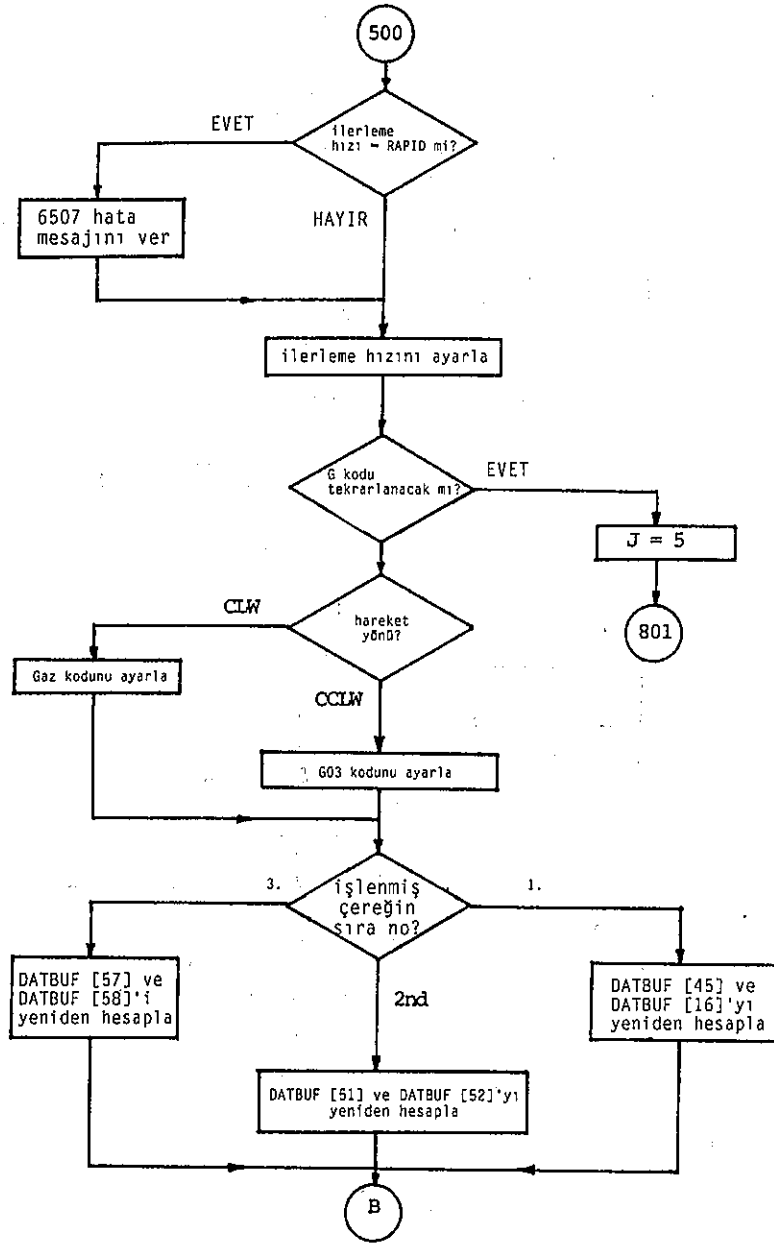
İlerleme hızı devirde inç cinsinden tanımlandığında özel bir deyim hareket



Şekil 15-8 (devamı var)

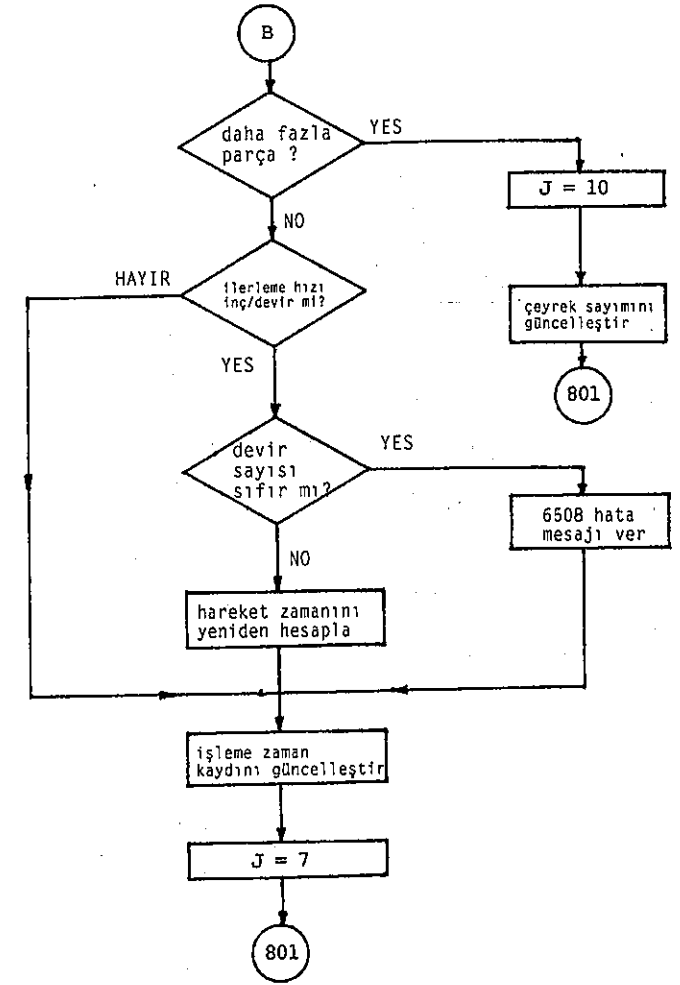
zamanını hesaplar. Bu program kısmı sonunda işleme zamanı güncelleştirilir. Devir sayısı sıfır olduğunda 6508 hata mesajı verilir.

Özel Fonksiyon Bölümü. Bu bölüm (Şekil 15-10) PREFUN'n APT deyimini tarafından tanımlanan G kodları ve F, M00, M01, M02, M08, M09 ve G04 kodlarını işler. PREFUN deyimini ile tanımlanan F kodu ve G kodunu ve END, OPSTOP, STOP ve DELAY deyimleri ile tanımlanan M ve G kodlarını işleyen üç kısma



Şekil 15-9 Şekil 15-1'de listelenen MTM'nin dairesel enterpolasyon hareket bölümü için akış diyagramı.

bölüme ayrılır. M02 kodu (program sonu kodu) türetildiğinde, özel bir işleme yordamı, iş mili ve soğutucunun durdurulmuş olduğundan emin olmak için prog-

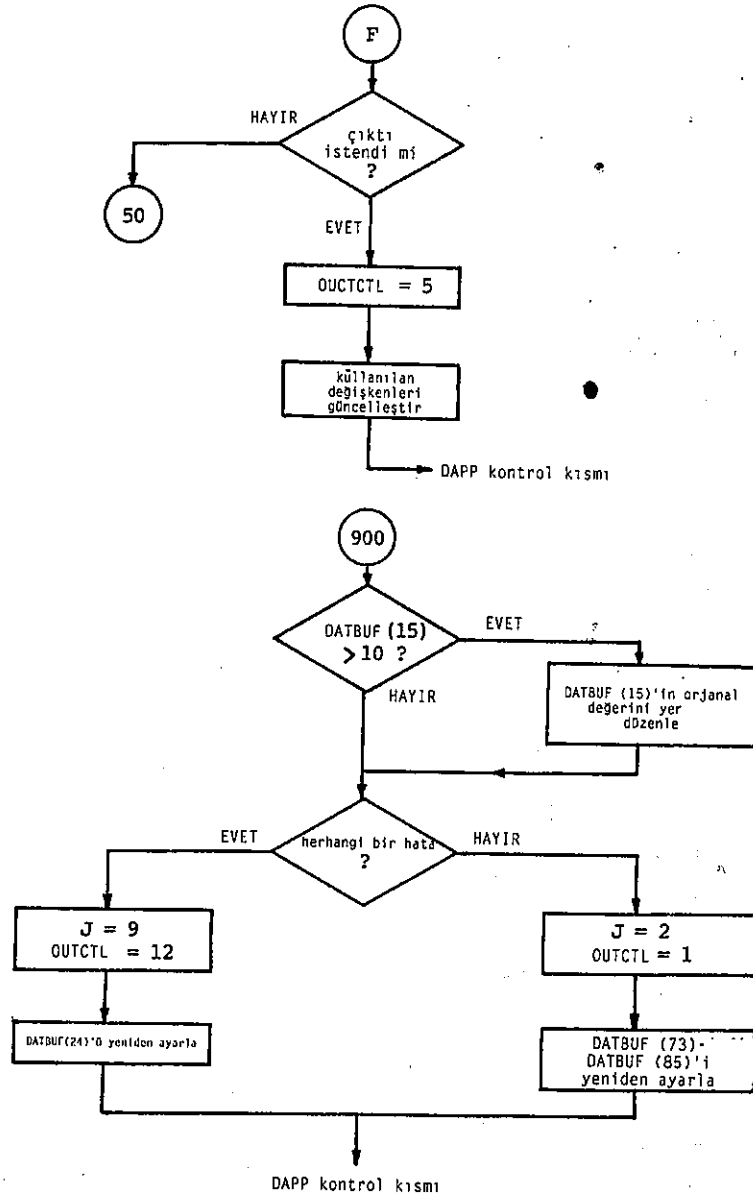


Şekil 15-9 (devamı var)

rama dahil edilmelidir. Eğer bir yardımcı fonksiyon (AUXFUN) yukarıda listelenenden başkasına programlanırsa 6503 hata mesajı verilir.

Hazırlama Bölümü. Önceki bölümler tarafından işlendikten sonra bir veya daha fazla kodlar bir blok olarak çıktıkları verilmeden önce, bunlar hazırlama bölümüne gönderilirler (Şekil 15-11). Bu bölüm tarafından ilgilenilmesi gereken bir kaç muhtemel durum vardır:

1. Eğer halihazırdaki bloкта çıktısı verilecek bir kod önceki bloktaki ile aynı ise, sadece bir tek blok içinde etkin olan kodlardan biri (örneğin, G04 ve



Şekil 15-12 Şekil 15-1'de listelenen MTM'nin denetim yönlendirme bölümü.

burada $IQ = -10000$ 'dir. Sonra hazırlama bölümünde tekrar 1'e dönüştürülür ve bir DATBUF elemanı $DATBUF(i)$ burada $73 \leq i \leq 85$ olarak saklanır.

Böylece örneğin, şunu hazırlama bölümünde tanımlayabiliriz:

$DATBUF(73)=1$

Son olarak, G01 kodunun çıktısı verilebilsin diye şunu ayarlarız:

$G=73$

- APT'deki x ve y koordinatları sırası ile z ve x koordinatlarına dönüştürülmelidirler. U kodu hariç olmak üzere, çıktılardan önce, (bekleme zamanını temsil eden) G04 kodu ile birlikte kullanılan X ve U kodları çap değerlerine dönüştürülmelidir (yani, mutlak ve artışı x koordinatlarının iki katı).

Kontrol Yönlendirme Bölümü. CLDATA grubunun parçası, hazırlama bölümünde hazırlandığında ve çıktısını almak için hazır olduğunda denetim, denetim-yönlendirme (1) bölümüne girer (Şekil 15-2, Şekil 15-1'e de bakınız). OUTCTL denetim yönlendirme değişkeni 5'e ayarlanır. Sonra denetim DAPP çıktı bölümüne gönderilir. İşlenen bilginin çıktısı alındıktan sonra, J değişkeni için ön ayarlama değeri tarafından yönlendirilen denetim MTM'nin işleme bölümüne geri döner. MTM işleme bölümü tarafından bütün okuma CLDATA grubu işlenene kadar bu çevrim tekrarlanır. Sonra, hazırlama bölümünün ilk deyimini (yani 800 etiketli deyim) icra edilebilir; bu J değişkenini 9'a ayarlar. İşleme verilerinin çıktıları alındıktan sonra, MTM'ye tekrar girdiğinde denetim doğrudan denetim yönlendirme (2) bölümüne daller (Şekil 15-1). Bu kısımda, eğer kesici kısmında değiştirildi ise $DATBUF(15)$ orjinal değerine geri döndürülür. $DATBUF(73)$ ile (85) arası elemanlar da yeniden ayarlanır. Eğer işleme hatası yoksa yeni bir CLDATA grubunu okumak için denetim DAPP'ın Ön İşlemler Bölümüne gönderilir. Aksi takdirde, denetim DAPP Ön İşlemler Bölümüne gönderilmeden önce bir hatanın olduğunu göstermek için bir satır yıldız yazılacaktır.

15.2.3 LTH2PP Son İşleminin ve İşleme Örneği Tasarlanması

Türetilen BLOCK DATA programı ve tasarlanan MTM'ü FORTRAN-H derleyici tarafından derlenir. Bunlar Bölüm 15-1'de listelenen EXEC'in icrası ile DAPP'la birlikte bağlantılı olarak düzeltilirler.

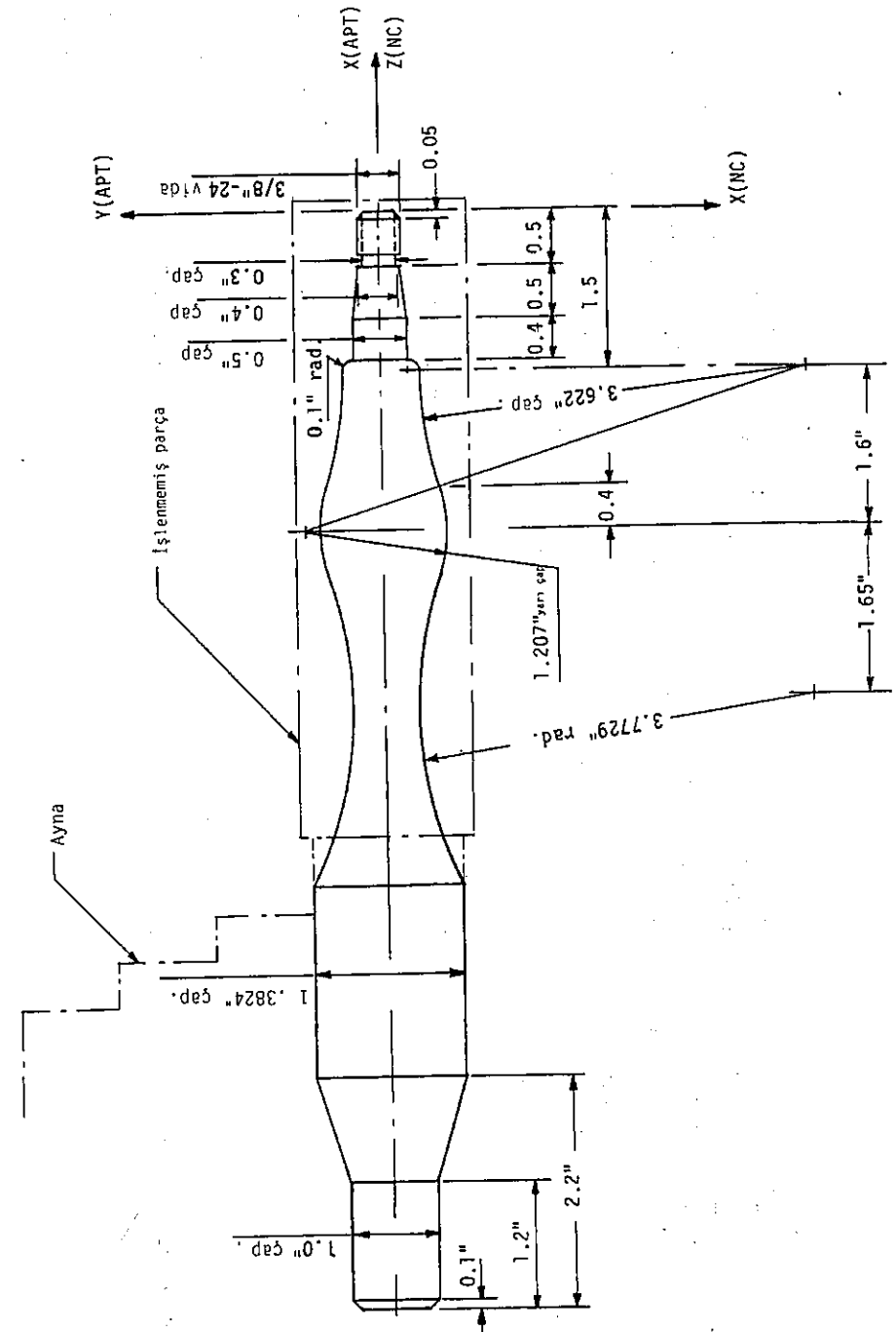
Bir örnek olarak, Şekil 15-13'de gösterilen parçayı işlemek için bir APT programı yazılmıştır. Bir önceki operasyonda milin bir kenarı ölçüsünde işlenmiştir. Şekil 15-14'deki APT programı, işleme operasyonunu sadece sağ kenar için

tanımlar. Bu programda sırası ile kaba talaş (takım No. 1), ince işleme (takım No. 2), kanal açma (takım No. 3) ve vida açma (takım No. 4) için dört kesici alet kullanılmaktadır. Bunlar döner başlıkta bu sırada saatin tersi yönünde monte edilmiştir. Koordinat sistemin orijini bitirilmiş uç yüzeyin merkezidir; bu koordinat sisteminde tezgah referans noktasının x ve y koordinatları sırası ile 12.5430 ve 7.6400'dür. LATH2PP son işlemci vida açma operasyonunu kullanmadığı için bu APT programında NC kodlarında vida açma çevrimini tanımlamak için bir INSERT deyimi kullanılır. Son işlemci listeleme çıktısı ve işlenmiş NC programı Şekil 15-15 ve 15-16'da verilmiştir.

DAPP yordamının kullanımı son işlemcinin tasarımını oldukça basitleştirir. Tasarımcı, DAPP'ta tezgahdan bağımsız işleme yordamı kullanabilir. Sadece tezgaha bağımlı işleme yordamı, NC denetimi sisteminin özellikleri ile ilgili veriler ve NC'li kod formatını sağlamak zorundadır. Fakat, eğer bir NC tezgah, vida açma çevrimi ve çoklu tornalama çevrimi gibi DAPP'ın kullanamayacağı fonksiyonlara sahip ise bazı DAPP yordamları (örneğin, DLWUP1 ve DLWUP2) düzenlenmelidir.⁷ Gerekli işleme fonksiyonunu gerçekleştirmek için MTM'de özel yordamlar da gerekebilir.

PROBLEMLER

- 15.1 Atölyenizde veya laboratuvarınızda bir NC'li freze tezgahını seçiniz ve Bölüm 15.1'de açıklanan yöntemleri takip ederek DAPP tabanlı bir son işlemci tasarlayınız. Tam bir tezgah özelliklerini anlama ve onun NC programlama kuralları gereklidir.
- 15.2 Üç eksenli bir CNC freze tezgahı saydam bir akrilik levha üzerine fiziksel bir nesnenin iki boyutlu görüntüsünü (fotoğrafını) yeniden üretmek için kullanılmaktadır. Siyah ve beyaz görüntü veya fotoğraf farklı bölgelerde değişen yoğunluklu beyaz bir zemin üzerine birçok siyah noktaların toplamıdır. Özel bir konumdaki siyah bir nokta "TRUE" durumu veya "1" rakamı ile ve beyaz olan "FALSE" durumu veya "0" rakamı ile temsil edilebilir. Siyah noktaların konumlarını temsil eden bir dizi koordinatlar nesnenin görüntüsünü sayısallaştırılmasından sonra elde edilebilir. Bu görüntü, her konumdaki siyah bir noktada küçük bir noktanın saydam akrilik levha üzerine oyulması ile yeniden üretilebilir. Levha üzerinde oyulan nokta beyaz bir nokta olur ve artık saydam değildir. Böylece bu nesnenin görüntüsü saydam levhanın siyah bir kağıt üzerine (siyah zemin) yerleştirilerek bir negatif kadar net olarak gösterilebilir. Aşağıdaki durumu farz ediniz:
- a. Görüntü sayısallaştırıcı tarafından işlendikten sonra, fiziksel nesnenin görüntüsü siyah noktalı konumları temsil eden bir grup koordinat tarafından temsil edilir (Şekil P15-2). Her koordinat grubu bir satır işgal eder.



Şekil 15-13 Vidalı bir profilin mil.

```

PARINO SAMPLE PROGRAM FOR CUTTING THE PART IN FIG. 15-13
MACHIN/LATH2,8,OPTION,2,12.5430,2
CUTCOM/7.64
P0=POINT/4.0,2.0
P1=POINT/0.05,0.8
P2=POINT/0.05,0.65
P3=POINT/-5.9,0.65
P4=POINT/-3.1,0.65
P5=POINT/0.05,0.45
P6=POINT/-4.75,0.4
P7=POINT/0.05,0.25
L1=LINE/YAXIS
L2=LINE/YAXIS
L3=LINE/(P8=POINT/-0.05,0.1875),ATANGL,-45
L4=LINE/P8,(-0.5,0.1875)
L5=LINE/(-0.5,0.1875),(-1.0,0.25)
L6=LINE/(-1.0,0.25),PARLEL,L2
L7=LINE/PARLEL,L2,YLARGE,0.6912
C1=CIRCLE/-1.5,0.25,0.1
C2=CIRCLE/-1.5,(0.25+0.1+3.622),3.622
C3=CIRCLE/CENTER,(-3.1,-0.5843),SMALL,TANTIO,C2
C4=CIRCLE/CENTER,(-4.75,4.1143),SMALL,TANTIO,C3
SPINDL/800,CCLW
COOLNT/ON
CUTTER/0.03
LOADTL/1 $$ THE FIRST TOOL IS THE ROUGHING TOOL
FROM/(POINT/12.5430,7.64)
RAPID; GOTO/P1
FEDRAT/0.005
GO/L1
TLLEFT,GOLFT/L1,PAST,L2
RAPID; GOTO/P2
GOTO/P3
RAPID; GOTO/P1
RAPID; GOTO/P5
GOTO/P4
GOTO/P6
GOTO/P3
RAPID; GOTO/P1
RAPID; GOTO/P7
GOTO/P4
RAPID; GOTO/P1
THICK/0.02
A1=MACRO
GO/L2,(PLANE/0,0,1,0),L1
TLRGT,GOBACK/L1,PAST,L3
GOLFT/L3,PAST,L4
GOLFT/L4,L5
GORGT/L5,PAST,L6
GOLFT/L6,TO,C1
ARCSLP/ON; GORGT/C1,TANTIO,C2
ARCSLP/ON; GOFWD/C2,TANTIO,C3
ARCSLP/ON; GOFWD/C3,TANTIO,C4
ARCSLP/ON; GOFWD/C4,PAST,L7
TERMAC
CALL/A1
RAPID; GOTO/P0
LOADTL/2 $$ PUT THE FINISHING TOOL IN POSITION
CUTTER/0.005
FEDRAT/0.003

```

Şekil 15-14 Şekil 15-13'de gösterilen parçanın sağ kenarını işlemek için APT programı.

```

THICK/0
SPINDL/1300,CCLW
CALL/A1
RAPID; GOTO/P0
LOADTL/3 $$ PUT THE GROOVING TOOL IN POSITION
RAPID
GOTO/(POINT/-0.5,0.25)
FEDRAT/0.002
SPINDL/400,CCLW
GOLTA/0,-0.1,0
RAPID; GOLTA/0,0.3,0
RAPID; GOTO/P0
LOADTL/4 $$ PUT THE THREADING TOOL IN POSITION
RAPID;GOTO/P7
INSERT G76X0.3209Z-0.42K0.0271D0.01E0.041667A60
RAPID;GOTO/P0
SPINDL/OFF
COOLNT/OFF
END
FINI

```

Şekil 15-14 (devamı var)

- Bir noktadan diğerine X veya Y yönündeki en küçük mesafe 0.02 inç'tir.
- b. Sivri uçlu yıldız şeklinde bir tungsten karpit akrilik levha üzerinde 0.01 inç derinliğinde küçük bir nokta delmek için kullanılmaktadır.
- Sayılaştırılan görüntüyü işlemek ve orijinal görüntüyü yeniden üretmek veya akrilik levha üzerinde büyütme veya küçültme maksadı ile bir SD programı üretmek için bir program gerekmektedir. FORTRAN veya PASCAL gibi herhangi bir yüksek seviyeli bir dilde yazılabilen işleme programının akış diyagramını çizin ve işleme prensiplerini açıklayınız.

```

ITEM S/370 NUMERICAL CONTROL PROCESSOR DESIGN AID FOR POSTPROCESSORS (DAPP) VERSION 1, MCD 1 DATE=87.146 0
INC-100 LAMH/FANUC-6T CONTROLLER SYSTEM BEGIN EXECUTION OF DAPP-BASED POSTPROCESSOR NAMED LAMH2, THE MM NUMBER IS 8 ...
0 SAMPLE PROGRAM FOR CUTTING THE PART IN FIG. 15-13 POSTPROCESSOR
ON G X Z I K U W F S Y H TIME 12/13/33 PAGE 1
15.2800 12.5430 .0000 .0000 X OLD= + 12.54300 Y OLD= + 7.64000
.05000 X INCH= + 6.84000 Z INCH= + .00000 FEDRAT= + 1.00000
600.00000 RAPID= + 2.00000 SPDR= + 800.00000 TOOLCD= + 1.00000
.00000 COORD= + 4.00000 ERROR= + .00000 SPEC OF= CIRCPL=
.00000 XCENTER= ZCENTER= ARC RAD= CUI J= DEF (75)=
CUI DV= CUI DW= CUI DX= DEF (72)= DEF (73)= DEF (74)= DEF (76)= DEF (77)= DEF (78)= DEF (79)=
0003 0004 0005 0006 00 1.6000 .0500 38 0100 42 03 09 05 02 00-09:10 00-09:11 00-09:00 00-09:00 00-09:01 00-09:04
THIS PART OF OUTPUT HAS BEEN DELETED.
*****
TOTAL MACHINING TIME (IN HR-MIN-SEC) = 09 05 02
TOTAL TAPE LENGTH (IN FEET) = 10.88
0 7 DIAGNOSTICS GENERATED DURING DAPP PROCESSING
0 POSTPROCESSOR TERMINATION.
** END OF POST PROCESSING **
SECTION 4 ELAPSED CPU TIME IN MIN/SEC IS 0000/00.9666
TOTAL PART PROGRAM CPU TIME IN MIN/SEC IS 0000/01.6799
**** END OF APT PROCESSING ****

```

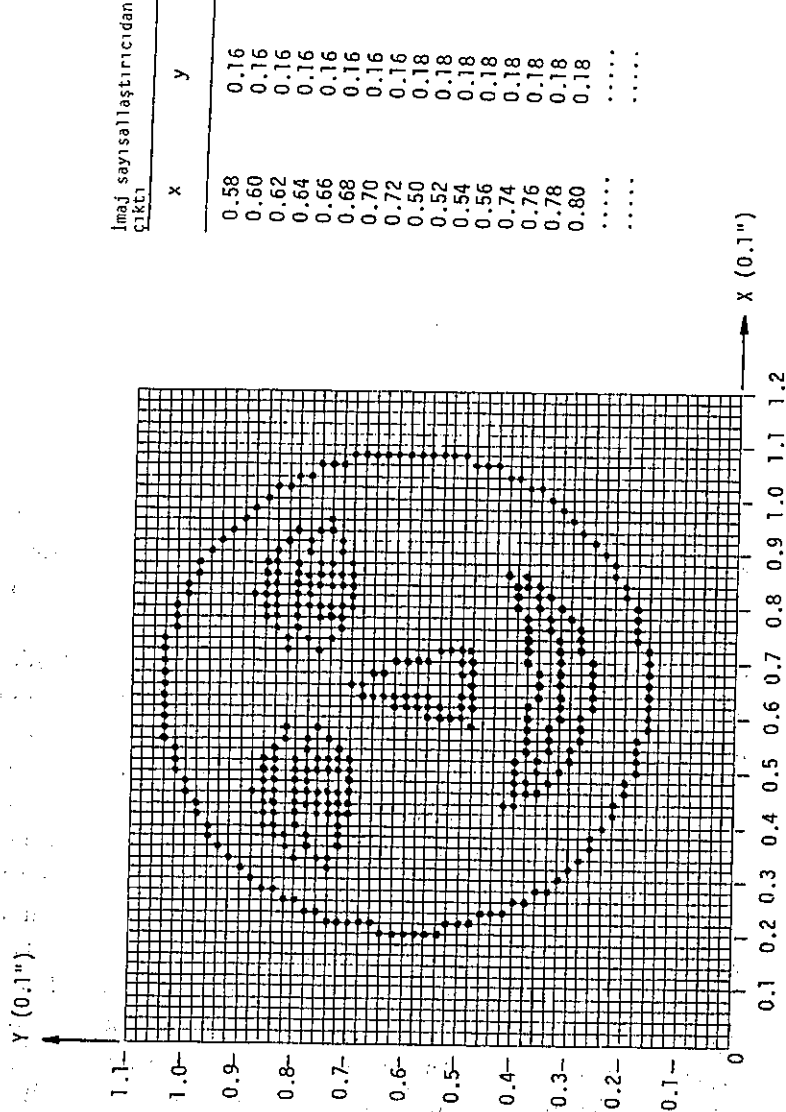
Şekil 15-15 Formata açıklama eklenmiş hali ile Şekil 15-14'de verilen APT programı için son işlemci çıktısı.

```

%*N0001G28U.0000W.0000*
N0002G50X15.2800Z12.5430*
N0003M08*
N0004S38T0100M42*
N0005M03*
N0006G00X1.6000Z.0500*
N0007G01Z.0150F.0050*
N0008X-.0300*
N0009G00X1.3000Z.0500*
N0010G01Z-5.9000*
N0011G00X1.6000Z.0500*
N0012G00X.9000*
N0013G01X1.3000Z-3.1000*
N0014X.8000Z-4.7500*
N0015X1.3000Z-5.9000*
N0016G00X1.6000Z.0500*
N0017G00X.5000*
N0018G01X1.3000Z-3.1000*
N0019G00X1.6000Z.0500*
N0020G01X.0700Z.0350*
N0021X.3040*
N0022X.4450Z-.0355*
N0023Z-.4978*
N0024X.5700Z-.9978*
N0025Z-1.3696*
N0026G03I-.0350K-.1304U.2000W-.1304*
N0027G02I3.5870U.4052W-1.1885*
N0028G03I-1.1719K-.4115U.1402W-.4115*
N0029I-1.2420U-.1402W-.4115*
N0030G02I3.5267K-1.2385U-.4222W-1.2385*
N0031I3.7378U.6994W-1.5787*
N0032G00X4.0000Z4.0000*
N0033T0200M11*
N0034G01X.0050Z.0025F.0030S64*
N0035X.2770*
N0036X.3800Z-.0490*
N0037Z-.4998*
N0038X.5050Z-.9998*
N0039Z-1.3975*
N0040G03I-.0025K-.1025U.2000W-.1025*
N0041G02I3.6195U.4088W-1.1992*
N0042G03I-1.1412K-.4008U.1366W-.4008*
N0043I-1.2095U-.1366W-.4008*
N0044G02I3.5574K-1.2492U-.4260W-1.2492*
N0045I3.7704U.6996W-1.5858*
N0046G00X4.0000Z4.0000*
N0047T0300M11*
N0048G00X.5000Z-.5000*
N0049G01X.3000F.0020S17*
N0050G00X.9000*
N0051G00X4.0000Z4.0000*
N0052T0400M11*
N0053G00X.5000Z.0500*
G76X0.3209Z-0.42K0.0271D0.01E0.041667A60*
N0054G00X4.0000Z4.0000*
N0055M09*
N0056M05*
N0057M02*

```

Şekil 15-16 Şekil 15-14'de listelenen APT programından türetilen NC programı.



Şekil P15-2

IV. Kısım İçin Son Söz

Bir son işlemciye olan ihtiyaç, kullanıcının benzer kapasiteli, fakat farklı tasarımlı NC'li tezgahlarda aynı parçayı işlemek için aynı NC programını kullanmasına engel olan denetimci tasarımının çeşitliliğinden ortaya çıkar. Onun için CLDATA'yı gereken NC kodlarına çevirmek için her tezgah için bir son işlemci gereklidir. Bir son işlemcinin türettiği NC programı taşınabilir değildir; bu sadece son işlemcinin kendisi için tasarlandığı tezgahla kullanılabilir. Onun için son işlemcinin kullanımı NC program taşınabilirlik problemini çözemez. Eğer aynı parça farklı tasarımlı iki NC'li tezgahta işlenecekse, her son işlemci tarafından bir kere, CLFILE iki kere işlenmelidir. Onun için, bu yaklaşım eldeki problemi çözmek niyeti ile değil aksine problemi ona yerleştirmeye yönelik olması yönü ile pasiftir.

Son işlemcilerin kullanımı bilgisayarda yükü iki yönde artırır, bunlar işleme zamanı ve boş hafızadır. Bir APT programının son işlemesi için maliyet (veya zaman) genellikle toplam işleme maliyetinin % 30 ve % 60'dır.¹² Bilgisayar kütüphanesinde tutulması gereken son işlemci sayısı NC'li tezgah sayısı tarafından tespit edilir. Bir atölyede ne kadar çok NC'li tezgah olursa (her birinin büyük bir dosyaya sahip olduğu) o kadar çok son işlemci ve o kadar çok ilgili problem sayısı olur. İmalat yönetimi ve imalatta meydana gelen zorluk ve karışıklık daha az esnek ve idare edilebilir olur.

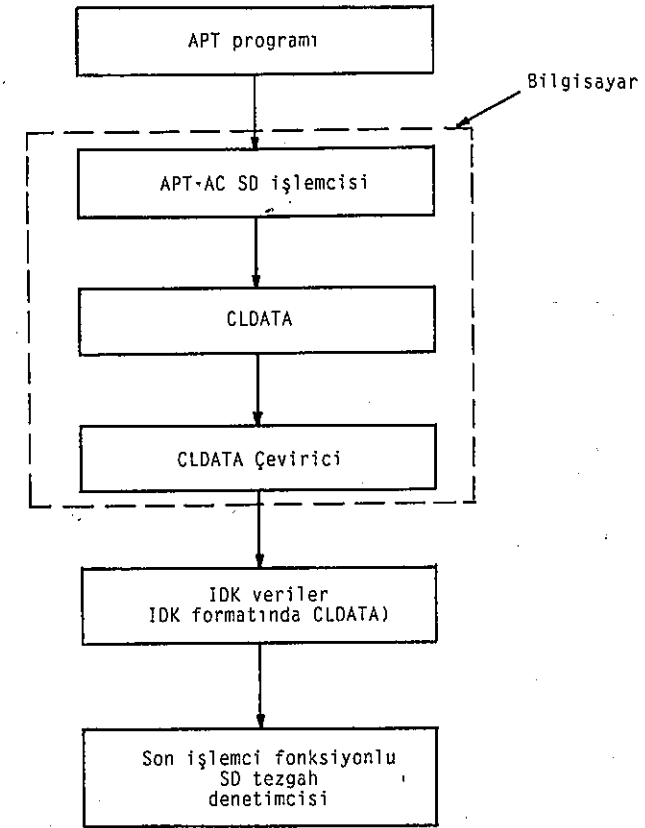
NC programlarının değiştirilebilirliğinin olmamasının neden olduğu problemler, birkaç NC'li tezgah, robot ve hareketli vasıtaların bir bilgisayarın kontrolü altında bütünleşmesinin yapıldığı Esnek Üretim Sistemi (FMS) gibi Bilgisayar Bütünlüklü Üretim Sistemlerinde (CIMS) daha kritik olur. CAD/CAM sistemlerinin fabrika cihazları ile bütünlüğü genellikle NC son işleme safhasında engellenmektedir.¹³ Kullanıcıların karşılaştığı engeller, son işlemcilerin alınması ve hata ayıklamak için gerekli zaman ve masraflardan eski model son işlemciler ile yeni CAD/CAM sistemleri ile çok sık olan uyumsuzluğa kadar uzanır.

Bu problemleri çözmek için kusursuz yol, NC denetimler için girdi formatını (NC kodları) standartlaştırmaktır. O zaman bir NC işlemcisi çıktısını standart NC kodlarında üretmek için tasarlanabilir. Sonuç olarak, son işlemciye gerek yoktur. O zaman bir NC programı farklı markadaki aynı tip NC'li tezgahlar arasında taşınabilir olur. Prensipite kusursuz olmasına rağmen, bu yaklaşımın gerçekleştirilmesi çok zordur, çünkü takım tezgahı imalatçıları kendi tasarım özelliklerini muhafaza etmeye meyillidirler. Çünkü kullanılan birçok standart olmayan NC'li tezgahlar vardır. NC denetimciler girdi formatı standartlaştırılmadığı sürece daima son işlemciler ihtiyacı vardır. Fakat, son işlemeyi başarmak için iki yol vardır:

1. Son işleme bilgisayar tarafından icra edilir.
2. Son işleme NC kontrolcu tarafından icra edilir.

Birincisi günümüzde uygulanan yaklaşımdır, bu önceki bölümlerde geniş olarak tartışıldı. İkinci yaklaşım CLDATA'yı NC'li tezgah denetimci sistemine girdi olarak kullanmadan ibarettir. NC denetimciler, CLDATA'yı kabul edip son işleme tabi tutabilmesi maksadı ile genişletilmiş fonksiyonları içermelidirler veya bunlarla uyumlu hale getirilmelidirler. Bu yaklaşım her nasılsa yukarıda bahsedilen fikir ile benzerdir, fakat bu daha gerçekçidir çünkü, günümüzdeki NC'li tezgahlar büyük değişiklikler olmaksızın uyumlu hale getirilebilir ve kullanılabilirler.

Güncel SD işlemci çıktısı, CLDATA işleri veya parçaya göre işlemi tanımlar. İçindeki bilgi tezgahla ilgili değildir. CLDATA bilgisayar için çağırması ve işlemesi kolay olan bir formattadır. CNC denetimcinin sınırlı işleme kabiliyetinden dolayı (CNC denetimcinin maliyetinin çok yüksek olacağı için aşırı işleme kabiliyeti ve fonksiyonları istenmez) NC denetimciler CLDATA girdisi basitleştirilmiştir. CLDATA'nın NC denetimciler girdi olarak kullanan temel kavramı önce 1974'te Rockwell International (El Segundo, Calif.) tarafından tanımlandı ve Vega Incorporated'in yardımı ile 1975'te uygulandı.^{12,13} FMS, CAD/CAM sistemleri ve CIMS'lerinin hızla yayılmasının neden olduğu ihtiyaçlardan dolayı Electronic Industries Association 1983'te NC'li tezgahlara girdi olarak kullanılan CLDATA için bir standart format geliştirdi. EIA RS-494 standardına göre¹⁴ kelimeler, sayılar ve karakterleri içine alan CLDATA'nın içeriği NC denetimci için işlemesi daha



SD tezgaha girdi olarak İDK verilerini kullanımını gösteren bir diyagram.

kolay olan 32-bit'lik ikili değişim kodu (İDK) veya kısaca İDK'li dosya bir parçaya bağlı olarak hareketleri tanımlar; böylece NC'li tezgah için işleme operasyonları tanımlayan tezgaha dayalı standart olan EIA RS-274-D ile karşılaştırıldığında EIA RS-494 parçaya dayalı bir standarttır. Burada gösterilen şekil NC denetimciye girdi olarak İDK kullanımı kavramının bir diyagramını takdim etmektedir.

Şekilden hemen görüleceği gibi NC'li tezgahlara girdi olarak İDK verilerinin kullanımı CLDATA'yı İDK standart formatına çevirmek için bir CLDATA çeviricisi (örneğin bir bilgisayar programı) gerekir, çünkü farklı NC işlemcileri farklı CLDATA kayıtlarının fiziksel temsiline sahiptirler. Çoğu NC'li tezgahlar, İDK verilerini girdi programlar olarak kabul edemediği için NC denetimcinin düzeltilmesi de gereklidir. Günümüzde, çok az NC'li tezgahlar EIA RS-494 standardına göre tasarlanmıştır. Standartı uyarlamak zamana, çalışmaya ve paraya büyük ölçüde yatırım demektir. İDK verisinin CNC denetimcisine girdi olarak kullanımının

avantajları ve dezavantajları tespit edilmelidir, çünkü EIA RS-494 standardı kullanıcılar ve takım tezgahı imalatçıları tarafından yaygın olarak kabul edilmemiştir.^{13,15} İDK veri girdisinin kullanımı, NC programlarını (deyim ve program formatı yönünden) aynı tip ve benzer kapasiteli tezgahlar arasında değiştirilebilir yapar, fakat, NC programlarının taşınabilirliğini engeller, NC programlarının taşınabilirliğindeki engeller sadece farklı NC'li tezgahlarda kullanılan NC'li tezgahların konfigürasyonuna, ulaşılabilir hassasiyete, rijitlik, tezgah gücü, devir sayısı ve ilerleme miktarındaki sınırlamalara en uygun kesicilere bağlı olarak bir tezgahtan diğerindeki farklı durumlardan kaynaklanabilir. İDK yaklaşımının bir şirkete faydalı olması diğerlerine de faydalı olacağı anlamına gelmez. Yukarıda takdim edilen analizden de anlaşılacağı üzere, İDK standardının asıl amacı, NC'li tezgahlar arasında parça programlarının taşınabilirliğini sağlamaktır. Bu yönde bir adım atmak bilgisayar ile NC'li tezgahlar arasında bir iletişim bağlantısının kurulması gerektiği anlamına gelir. Bir bilgisayar otomasyonlu üretim sistemi veya doğrudan veya dağıtılmış sayısal denetimli bir grup NC'li tezgahlar İDK verilerinin girdi olarak kullanımından fayda görebilirler. Fakat, bilgisayarlar ve NC'li tezgahlar arasında bağlantısız her tipte sadece bir NC'li tezgaha sahip olan bir şirket sadece İDK kullanımının hatırı için uyarlamayı haklı çıkarmakta zorluk çekecektir. İDK'nın uyarlanması artık son işlemcilerle ihtiyacımız olmadığı anlamına gelmediğine dikkat edilmelidir. Gerçekte, İDK verisi için hala son işlemci gerekir, fakat son işlemci bilgisayardan ayrılır ve CNC denetimciye ilave edilir.

Bu arada, birçok NC yazılım satıcıları son işlemcilerin tasarımını, NC'li tezgah ile ilgili bir seri soruların cevaplarına onların tasarımını azaltarak basitleştirmeye çalışmaktadırlar.¹⁶ Eğer bu cevaplar hazırlanırsa, özel bir NC'li tezgah için isteğe göre son işlemci türetilebilir. Bu yaklaşımın gerçekleştirilmesi, son işlemcinin türetilmesi için bir kaç işleme programlarının yazılıma eklenmesini gerektirir. Bu yazılımın çok yönlülüğü, ilave edilen işleme yordamlarının sayısı ve bu sorulara cevapların kolaylığı tarafından tespit edilir. Birbirleri ile ilişkilerinin bulunması, sayılarının fazla olması, bazılarının anlaşılmasının zorluğu ve belirsizliğinden dolayı bu gibi yazılımın kullanımındaki ana problemlerden birisi bu sorulara doğru olarak cevaplamaktır. Diğer bir problem ise, işleme yordamlarının bütün tezgah tiplerinin fonksiyonlarını kapsamasıdır. Kullanıcının hala kendi tezgahı için özel olarak bir veya daha fazla alt program tasarlama gerekir. Ayrıca, eğer son işlemci üretici yazılımındaki işleme yordamı, özel bir tezgahın fonksiyonu için gerçek ihtiyaçları karşılamazsa, genellikle yeni bir kullanıcı bunun düzeltilmesinde oldukça fazla zorlukla karşılaşır.

Özet olarak, kullanıcının bu soruları doğru olarak anlaması, cevaplaması ve yazılıma eklenen işleme yordamlarının bu özel tezgahın tüm fonksiyonları içermesi şartı ile bu yazılım, az bir tecrübeli bir kişi tarafından isteğe göre bir son işlemci türetmek için son işlemci tasarımında kullanılabilir.

NC işleme programlarının taşınabilirliğini iyileştirmek ve CAD/CAM sistemleri ve NC'li tezgahlar arasındaki arabirim için başka bir yol var mı? Kapasitedeki artışla beraber, boyutta ve bilgisayar maliyetindeki düşüşten dolayı fiziksel bağlantılı NC sistemden daha fazla fonksiyonlara sahip ve kullanmak için daha elverişli olan Bilgisayarlı Sayısal Denetim (CNC) sistemi NC'nin gelişmiş halini temsil eder. Bilgisayarların boyut ve maliyetinde düşüş eğilimi devam ettiği için CNC denetimcilerinin daha güçlü amaca-özel bilgisayarları içereceğini tahmin edebiliriz. Etkileşimli NC programlama ve kesici yolunun grafik gösterimi birçok CNC tezgahlarında gerçekleştirilmiştir. Bir son işlemcinin fonksiyonları da yukarıda açıklandığı gibi eklenebilir. APT programını NC denetimcisine girdi olarak kullanabilmeyi basitleştirmek için NC işlemcisinin fonksiyonlarını da bir tezgaha ekleyebilir miyiz?* Bilgisayar teknolojisindeki gelişmenin ışığı altında, bu amaca ulaşılabilirliğini ve buna ulaşmanın sadece zaman ve gayret meselesi olduğunu düşünüyoruz. Takım tezgahı imalatçıları arasındaki yarış ve böylece NC denetimci tasarımındaki büyük farklılık geçmişte NC teknolojisinin hızlı gelişimini sağladı; fakat bunlar bugünkü bilgisayar bütünlüklü üretim teknolojisinin gelişmesine bir engel oldu. Birçok alanda endüstri tarihinin gösterdiği gibi, tasarımın standartlaştırılması bilgisayar bütünlüklü üretim sistemlerinin gerçekleştirilmesinde para, zaman ve gayretten kar etmenin en iyi yoludur.

IV. PARÇA İÇİN KAYNAKLAR

1. ISO Standard (ISO 6983/1-1982): *Numerical Control of Machines: Program Format and Definition of Address Words. Part 1: Data Format for Positioning, Line Motion and Contouring Control Systems*, 1st ed. Geneva, Switzerland: International Organization for Standardization, 1982.
2. EIA Standard (RS-274-D): *Interchangeable Variable Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically Controlled Machines*. Washington, D.C.: Electronic Industries Association, 1979.
3. ISO Standard (ISO 3592-1978): *Numerical Control of Machines: NC Processor Output - Logical Structure and Major Words*. Geneva, Switzerland: International Organization for Standardization, 1978.
4. ISO Standard (ISO 4343-1978): *Numerical Control of Machine: NC Processor Output - Minor Elements of 2000 Type Records (Postprocessor Commands)*. Geneva, Switzerland: International Organization for Standardization, 1978.

* Bu kavram özel bir sohbet esnasında IBM şirketinden Dr. David Grossman tarafından önerildi.

5. Subramanian, M. L., "CLFILE Manipulator," *Advancing Manufacturing Technologies, Proceedings of the 21st Annual Meeting and Technical Conference*, Numerical Control Society, 1984, pp. 24963.
6. IBM Manual (SH20-1414-2): *Automatically Programmed Tool Advanced Contouring Numerical Control Processor: Program Reference Manual*, 3rd ed. Rye Brook, N.Y.: IBM Corp., 1985.
7. IBM Manual (SH20-2469-0): *System/370 Automatically Programmed Tool - Intermediate Contouring and Advanced Contouring Numerical Control Processor: Program Reference Manual*. Rye Brook, N.Y.: IBM Corp., 1980.
8. Sim, R? M., "Postprocessor," *Numerical Control User's Handbook*, W. H. P. Leslie, ed. New York: McGraw-Hill, 1970, pp. 299-344.
9. Chang, Chao-Hwa, "A Postprocessor for Computer-Aided Programming of the NC-100 Lathe/GN6T Controller systems based on the IBM System/370 APT-AC NC Processor," Master's degree thesis, University of California, Los Angeles, 1983.
10. Patton, W. J., *Numerical Control: Practice and Application*. Reston, Va.: Reston Publishing, 1972.
11. IBM Manual (SH20-1413-2): *System/370 Automatically Programmed Tool - Advanced Contouring Numerical Control Processor: Operations Guide*, 3rd ed. Rye Brook, N.Y.: IBM Corp., 1982.
12. Herndon, L. R., Jr., "CLdata Input for CNC: The End of the Line for Postprocessor," *NC - From the User's Point of View*, Numerical Control Society, 1977.
13. Ogorch, M., "CNC Standard Format," *Manufacturing Engineering*, Jan. 1985, pp. 43-5.
14. EIA Standard (RS-494): *32 Bit Binary CL Exchange (BCL) Input Format for Numerical-Controlled Machines*. Washington, D.C.: Electronic Industries Association, Aug. 1983.
15. Justice, R. K., "Capture the BCL Concept for Present Day Controllers," *Advancing Manufacturing Technologies, Proceedings of the 21st Annual Meeting and Technical Conference*, Numerical Control Society, 1984, pp. 241-8.
16. IBM Manual (GH20-6908-0): *Numerical Control Postprocessor Generator: General Information*, 1st ed. Rye Brook, N.Y.: IBM Corp., 1986.

Ekler

EK A APT DİLİNDE KULLANILAN KELİMELEER

AAXIS	ABSF	ABSLTE	ACOSF
ADD	ADJUST	AIR	ALL
ALPNUM	ANGLE	ANGLF	ANGTOL
ANTSPI	APT360	ARC	ARCSLP
ASINF	ASLOPE	AT	ATANF
ATANGL	ATANZF	ATTACH	AUTO
AUTOPS	AUTPOL	AUXFUN	AVOID
AXIS	BACK	BAXIS	BCD
BCHIP	BEVEL	BEVELS	BINARY
BISECT	BLACK	BLANK	BLUE
BORE	BOREOS	BOTH	BREAK
BRKCHP	CALL	CAM	CAMERA
CANON	CATLOG	CAXIS	CBORE
CERTF	CCLW	CENTER	CHANGE
CHECK	CHORD	CHUCK	CIRCLE
CIRCUL	CIRLIN	CLAMP	CLDIST
CLEARP	CLEARV	CLFILE	CLPRNT
CLRSRF	CLTV	CLW	CM
CMIT	CNSINK	COLLET	COMBIN
CONE	CONSEC	CONST	CONT
CONTIN	CONTUR	COOLNT	COPY
CORNFD	COSF	COTANF	COUPLE
CROSS	CRSSPL	CS	CSINK
CTREAC	CURSEG	CUT	CUTANG
CUTCOM	CUTTER	CYCLE	CYLNDR
DAC	DARK	DASH	DATA
DATREF	DCOORD	DEBUG	DECR
DEEP	DEEPL	DELAY	DELET
DELETE	DELTA	DEPTHV	DHOLE
DIAG1	DIAG2	DIAG3	DIAMTR
DISPLY	DISTF	DITTO	DMILL
DNTCUT	DNTLR	DNTLRP	DNTR
DO	DOT	DOTTED	DOWN
DRAFT	DRAWLI	DRESS	DRILL
DS	DSTAN	DWELL	DWELLV
DWL	DYNDMP	EDIT	EDITND
ELLIPS	ELMSRF	END	ENDARC
ERCOND	EXEC	EXPF	FACE
FACEML	FAN	FEDRAT	FEDTAB
FEED	FEEDRT	FEET	FILE
FINI	FINISH	FIX	FLOOD
FLOW	FMATL	FORMAT	FOURPT
FREE	FROM	FRONT	FULL
UNOFY	GAPLES	GCONIC	GENCUR
	GOBACK	GOCLER	GODLTA
	GOFWD	GOHOME	GOLFT
	GOTO	GOOGCK	GOUP
	GRID	GROOVE	HEAD
	HOLDER	HYPERB	ICODEF

Automatically Programmed Tool Advanced Contouring (APT-AC) Numerical Reference Manual, 3rd ed. Rye Brook, N.Y.: IBM Corp., 1985. (Courtesy of IBM Corporation.)

IF	IFRO	IN	INCHES
INCR	INDEX	INDIRP	INDIRV
INDNRM	INDVEC	INSERT	INTCOD
INTEGR	INTENS	INTERC	INTGF
INTGRV	INTOF	INTOL	INVERS
INVOC	INVX	INVY	IPM
IPR	ISTOP	JUMPTO	KEYBOR
LARGE	LAST	LATER	LCOMC
LEADER	LEFT	LENGTH	LETTER
LIBRY	LIFTOF	LIGHT	LIMIT
LIMSRF	LINCIR	LINE	LINEAR
LINK	LINTOL	LIST	LITE
LINTHF	LOADTL	LOCAL	LOCK
LOCKX	LOFT	LOGF	LOG10F
LOOPND	LOOPST	LOW	LPRINT
LTV	MACH	MACHIN	MACRO
MAGTAP	MAIN	MAJOR	MANUAL
MATL	MATRIX	MAXDP	MAXDPM
MAXIPM	MAXRPM	MAXVEL	MAXIF
MCHFIN	MCHTOL	MDEND	MDWRIT
MED	MEDIUM	MESH	MILL
MINOR	MINUS	MINIF	MIRROR
MIST	MIT	MM	MMPM
MMPR	MODE	MODF	MODIFY
MOTION	MOVETO	MULTAX	MULTRD
MXMMPM	NAUDIT	NCDB	NCTEST
NEGX	NEGY	NEGZ	NEXT
NIXIE	NOCS	NOMORE	NOPLT
OPOST	NOPS	NORMAL	NORMDS
ORMPS	NOSLP	NOW	NOX
NOY	NOZ	NUMBER	NUMF
NUMPTS	OBTAIN	OFF	OFFSET
OMIT	ON	OPEN	OPRINT
OPSKIP	OPSTOP	OPTION	ORIGIN
OUT	OUTIOL	OVCONT	OVPLT
PARAB	PARAM	PARLEL	PART
PARTNO	PASS	PAST	PATCH
PATERN	PBS	PEN	PENDWN
PENUP	PERPTO	PERSP	PI
PICKUP	PILOTD	PITCH	PIVOTZ
PLABEL	PLANE	PLOT	PLUNGE
PLUS	PNTSON	PNTVCT	PNTVEC
POCKET	POINT	POLCON	POLYGN
POSMAP	POSTN	POSX	POSY
POSZ	POWER	PPLOT	PPRINT
PPWORD	PREFUN	PRINT	PROBX
PROBY	PROC	PROCND	PROPF
PS	PSIS	PSTAN	PTFORM
PTNORM	PTONLY	PTSLOP	PULBOR
PULFAC	PUNCH	QADRIC	QUILL
RADIUS	RAIL	RANDOM	RANGE
RAPID	READ	REAM	REAMA
REAR	RED	REDEF	REFSYS

(devamı var)

REGBRK	REMARK	REPLAC	RESERV
RESET	RETAIN	RETRCT	REV
REVERS	REVOLV	REWIND	RIGHT
RLDSRF	ROOT	ROTABL	ROTHED
ROTREF	ROUGH	ROUND	RPM
RTHETA	RULED	SADDLE	SAFETY
SAME	SCALE	SCRIBE	SCRUCT
SCULPT	SCURV	SECTN3	SEC1
SEC2	SEC3	SEG	SELCTL
SEQNO	SETANG	SETOOL	SFM
SIDE	SIGNF	SINF	SLOPE
SLOWDN	SMALL	SMESH	SOLID
SOURCE	SPALIN	SPDRL	SPDTAB
SPECDP	SPECFR	SPEED	SPHERE
SPINDL	SPINSP	SPLINE	SPMIL
SORTF	SRFREV	SEFVCT	SSURF
START	STEP	STOP	SWITCH
SYN	SYSLIB	TABCYL	TABPRT
TANDS	TANF	TANON	TANSPL
TANTO	TAP	TAPKUL	TERMAC
TEXT	THETAR	THICK	THREAD
THRU	TIME	TIMES	TITLES
TLAXIS	TLIFT	TLDON	TLOFPS
TLOXIS	TLOPNS	TLRGT	TMARK
TO	TOLER	TOOL	TOOLNO
TOOLST	TORS	TORUS	TP
TPI	TPMM	TRACUT	TRANPT
TRANS	TRANSL	TRANTO	TRAV
TRFORM	TRMCOD	TRYBOR	TRYBOS
TUNEUP	TURN	TURRET	TWOPT
TYPE	TYPEF	UAXIS	ULOCKX
UNIT	UNITS	UNLIKE	UNLOAD
UP	VAXIS	VECTOR	VTLAXS
WAXIS	WCORN	WDEFAC	WEIGHT
XAXIS	XCOORD	XLARGE	XREF
XSMALL	XYOR	XYPLAN	XYROT
XYVIEW	XYZ	YAXIS	YCOORD
YLARGE	YSMALL	YZPLAN	YZROT
YZVIEW	ZAXIS	ZCOORD	ZERO
ZICZAG	ZLARGE	ZSMALL	ZSURF
ZXPLAN	ZXROT	ZXVIEW	ZDCALC
ZDCALC	3PT2SL	4PT1SL	5PT

EK B APT'DE TANIMLANAN SEÇİLMİŞ GEOMETRİK ÖGELERİN KURALINA UYGUN BİÇİMLERİ

Geometrik Ögenin Tipi	Hafızada Saklama Tarihi veya Listeleme Dosyasındaki Görünüş	Açıklama
POINT	X,Y,Z	X,Y,Z seçilen koordinat sisteminde noktanın kartezyen koordinatları.
LINE	A,B,C,D	A,B,C ve D normalleştirilmiş doğru denkleminin sabitleridirler: $AX + BY + CZ - D = 0$ APT'de bir doğru X-Y düzlemine dik bir düzlem olarak işlem gördüğü için C daima sıfırdır.
CIRCLE	X,Y,Z,A,B,C,R	X, Y ve Z merkez noktasının koordinatlarıdır: R yarıçaptır. Bir çember Z eksenine paralel bir silindir olarak işlem görür. A, B ve C sırası ile silindir eksenine paralel birim vektörünün X, Y ve Z bileşenleridir. Böylece A, B ve C sırası ile 0, 0 ve 1'dir.
VECTOR	A, B, C	A, B ve C sırası ile vektörün X, Y ve Z bileşenleridir.
PLANE	A,B,C,D	A, B, C ve D düzlem denkleminin sabitleridirler: $AX + BY + CZ - D = 0$
CYLNDR	X,Y,Z,A,B,C,R	X, Y ve Z silindir ekseninde tanımlanan noktanın koordinatlarıdır. A, B ve C sırası ile eksenel birim vektörünün X, Y ve Z bileşenleridir. R yarıçaptır.
QUADRIC	A,B,C,D,F,G,H,P,Q,R	Bunlar genel çeyrek yüzey denkleminin sabitleridirler: $AX^2 + BY^2 + CZ^2 + D + 2FYZ + 2GXZ + 2HXY + 2PX + 2QY + 2RZ = 0$
GCONIC ve LCONIC	A ₁ ,A ₂ ,...,A ₁₀ ,C ₁ ,C ₂ ,...,C ₆ , M ₁ ,M ₂ ,...,M ₁₂ ,N ₁ ,N ₂ ,...,N ₁₂	A ₁ ile A ₁₀ arası çeyrek yüzey denkleminin sabitleri: $A_1x^2 + A_2y^2 + A_3z^2 + A_4 + 2A_5yz + 2A_6xz + 2A_7xy + 2A_8x + 2A_9y + 2A_{10}z = 0$ C ₁ ile C ₆ arası konik denkleminin sabitleridir: $C_1x^2 + C_2xy + C_3y^2 + C_4x + C_5y + C_6 = 0$ M ₁ ve M ₁₂ arası yukarıda gösterilen konik denklemi ile koninin tanımlandığı universal koordinat sisteminde bir noktayı aktaran ters dönüşümü tanımlar. N ₁ ile N ₁₂ arası M'in ters matrisinin sabitleridir.
MATRIX	M ₁ ,M ₂ ,...,M ₁₂	Bunlar dönüşüm matrisinin sabitleridir:

$$\begin{bmatrix} M1 & M2 & M3 & M4 \\ M5 & M6 & M7 & M8 \\ M9 & M10 & M11 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Değişiklik yapılarak alındığı yer IBM Manual SH20-1414-2: *Automatically Programmed Tool - Advanced Contouring (APT-AC) Numerical Control Processor: Program Reference Manual*, 3rd ed., Rye Brook, N.Y.: IBM Corp., 1985. (Courtesy of International Business Machines Corporation.)

tanımlanmadığı için $U-V-W$ koordinat sistemi $X-Y-Z$ parça koordinat sistemi ile aynıdır. İki nokta tanımlanan doğrultmanın uzantıları olarak eklenir, bunlar PT0 (5,49871750, -2,50170961) ve PT5 (19,9507012,8.45925412). Kolayca doğrulanabileceği gibi, bu uzatılmış doğruların uzunlukları, PT0-PT1 ve PT4-PT5, 10 birimdir.

Doğrultman $n + 1$ kısma bölünür, burada n TABCYL deyiminde tanımlanan noktaların sayısıdır. İlk ve son kısımlar uzatılmış doğrulardır. Aşağıdaki ifadeli bir parametrik kübik denklem her bitişik tanımlanan iki nokta arasındaki eğri kısımları tanımlamak için kullanılır:

$$t = As^3 + Bs^2 + Cs$$

burada

- A ve B = III. kısımdaki A ve B sütunlarında listelenen parametreler
 C = $-(AL^2 + BL)$, LENGTH sütununda listelenen L eğri kısmın iki uç noktaları arasındaki mesafeye eşittir
 s = $t-s$ koordinat sistemindeki koordinat. Her eğri kısmında, s eksenine başlama noktasından bitiş noktası yönünde olduğuna dikkat ediniz; böylece $0 \leq s \leq L$.

Örneğin, Şekil C-2'de listelenen çıktıya göre, PT2-PT3 eğri kısmı şu denklemle tanımlanır:

$$t = 0.023265677s^3 - 0.200997302s^2 + 0.432178776s$$

Her eğri kısım için t 'nin en büyük ve en küçük değeri L 'yi sırası ile MAX ve MIN sütunlarında listelenen parametre ile çarpılarak tespit edilebilir. Bir eğri içinde bir dönüm noktası varsa, en büyük ve en küçüğün ikisi de listede gözüktür; aksi takdirde sadece bir en büyük veya bir en küçük listelenir (diğeri sıfır olarak yazılır).

EK D

DAPP TABANLI SON İŞLEMCİNİN OLUŞUMU VE OS/MVS ORTAMINDA BİR APT PROGRAMININ İŞLENMESİ İÇİN İŞ DENETİM DİLİ (IDD) YAZILIMLARI

1. BLOCK DATA programını üretmek için İDD deyimleri:

```
//IS79CHC1 JOB IS79CHC,MSGCLASS=A,TIME=(,20)
// EXEC PGM=QUEST
//STEPLIB DD DSN=SYS1.DKWLM.PP,UNIT=3330,VOL=SER=NCPACK,
DISP=SHR
//FT06F001 DD SYSOUT=A
//FT07F001 DD SYSOUT=B
//FT08F001 DD UNIT=3330,SPACE=(TRK,4),
DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)
//FT05F001 DD *
.....
..... (DAPP sorularına cevaplar)
.....
/*
```

2. Bir son işlemci tasarımı için İDD deyimleri:

```
//IS79CHC2 JOB IS79CHC
// EXEC FORTHCL,PARM,FORT='OPT=2'
//FORT.SYSIN DD DSN=IS79CHC.COMBLO8,DISP=SHR
DD DSN=IS79CHC.MTMD08,DISP=SHR
/*
//LKED.SYSLMOD DD DSN=IS79CHC.POSTP8,DISP=(NEW,KEEP),UNIT=3330,
// VOL=SER=NCPACK,SPACE=(CYL,(5,1,5))
//LKED.OLDLIB DD DSN=SYS1.DKWLM.PP,UNIT=3330,DISP=SHR,
VOL=SER=NCPACK
//LKED.SYSIN DD *
INCLUDE OLDLIB (ZDAPPPP)
ENTRY DAPPPP
NAME LATH2PP
/*
```

burada BLOCK DATA programı ve Takım Tezgahı Modülü için veri grup numaraları sırası ile COMBLO8 ve TTMD08'dir. Tasarlanan son işlemci için veri grup ismi POSTP8'dir. İkinci son kartta (veya satırda), LATH2, E01 sorusunun cevabı tarafından tanımlanan son işlemcinin adıdır.

II. parça için 2. ve 8. referanslardan ve IV. parça için 9. referanstan alınarak değişiklik yapılmıştır. İDD deyimlerinde, IS79CHC farz edilen kullanıcı hesap ismi ve IS79CHCn, iş adıdır. Okuyucu veri grup ismini ve parametreleri değiştirmelidir ve kendi bilgisayar sisteminin ihtiyacına göre çıktıyı yazmalıdır.

3. Bir APT programını işlemek için İDD deyimleri:

```
//IS79CHC3 JOB IS79CHC
//GO EXEC PGM=DKW0,PARM='POOLSIZE=(10,5),LINECNT=76'
//STEPLIB DD DSN=SYS1.DKWLM.PP,UNIT=SYSDA,VOL=SER=NCPACK,DISP=SHR
DD DSN=IS79CHC.POSTP8,UNIT=SYSDA,VOL=SER=NCPACK,DISP=SHR
//ALINKLIB DD DSN=SYS1.DKWLM.PP,UNIT=SYSDA,VOL=SER=NCPACK,DISP=SHR
//ERRFILE DD SYSOUT=A
//FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=133,BLKSIZE=1995)
//DPOOL DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//LIBDPOOL DD DSN=APTAC.LIBDPOOL,UNIT=3330,VOL=SER=NCPACK,DISP=SHR
//SOURCE DD DUMMY
//SYSPUNCH DD SYSOUT=B
//POSPUNCH DD SYSOUT=B
//FT07F001 DD SYSOUT=B
//FT05F001 DD DSN=IS79CHC.PTPGM1,DISP=SHR
//
```

burada PTPGM1 işlenecek APT programıdır, POSTP8 kullanılan son işlemcinin veri grup adı; ve LIBDPOOL SD işlemci teşhis tablosunun veri grup adıdır.

İNDEKS

- 3PT2SL ikincil kelimesi, 167, 389
- 4PT1SL değiştiricisi, 166
- 5PT değiştiricisi, 166, 389
- ABSF fonksiyonu, 207
- ABSLTE ikincil kelime, 100,232
- ACOSF fonksiyonu, 207
- ADAPT, 10
- Adaptiv denetim, 12
- Affine transformation, 341
- ALL değiştiricisi, 285
- Altprogram, APT, 217, 332, 333
 - tanımlama deyimleri, 332
 - MACRO deyimini, 332
 - TERMAC deyimini, 332
 - örnekler, 333, 333, 337, 338
 - icra deyimini (CALL), 332
 - parametreler (değişkenler), 446
 - programlama hataları, 333, 337
 - programlama kuralları, 333
 - deyim sınırlaması, 333
 - hesaplama deyimleriyle kullanımı, 337
- Altprogram SD kodları, 67-68
- ANSI standartları
 - B29.1-1975, 216
 - X3.37-1980, 10, 93
 - Y14.5M-1982, 93
- ANULL, 483
- APT III Programı (işlemci), 9, 429
- APT matematiksel fonksiyonlar (tablo), 207-209
- APT uzun Alan Programı, 10
- APT (Otomatik Programlanmış Alet) dili, 9,
 - 95, 420
 - karakterler, 98, 99
 - tanımlama kesici yolu, 226, 393, 429, 349,
 - 352, 359, 365, 377
 - geometrik öğeler, 114, 120, 205
 - 328, 349, 352, 629
 - işleme özellikleri, 292, 309
 - delimiters, 98, 99, 103, 110
 - elemanlar, 98
- hata mesajları, 98
- torna tezgahı programlaması, 304, 384,
 - 395, 396
- freze tezgahı programlama, 387, 395
- sayılar, 98, 102, 103
- çıktı listesi, 408, 412
- program hata ayıklama, 383, 404, 408
- örnekler, 387, 397, 611
- oluşum, 383, 397, 402
- yapı, 385, 387
- programlama işlem sırası, 383, 387
- noktalama, 98, 99, 103, 110
- skalar, 102
- standart, 10, 93
- deyim formatı, 112
- deyim etiketi, 98, 110
- deyim boyutu, 112
- sembol, 98, 102, 110
- kelimeler (tablo), 98, 99, 102, 626
- esas, 102
- ikincil, tali, 102
- son işlemci, 99, 385,
- APT-AC SD işlemcisi (IBM APT-AC SD
 - işlemcisine bakınız)
- Ara (Beklemeye bakınız)
- ARC ikincil kelimesi, 122, 233
- ARCSLP deyimini, 307, 405, 468, 486,
 - 520, 534
- Aritmetik operatörler, 408, 109
- Artı işareti, 108
- ASCII kodu, 21
- ASINF fonksiyonu, 207
- ASSEMBLER dili, 477
- AT değiştiricisi, 230
- ATAN2F fonksiyonu, 207
- ATANF fonksiyonu, 207
- ATANGL değiştiricisi, 122, 157, 233, 517
- ATTACH değiştiricisi, 240, 244
- AUTOPS deyimini, 272
- AUXFUN deyimini, 480, 520, 574, 602
- AVOID değiştiricisi, 246, 249, 252, 535

- Ayna görüntüsü, 344, 345
Bağlantılı operatör (tablo), 209, 319
Baud değeri, 459
Başlangıç hareketi, APT, 264, 265, 271, 345
366, 369, 370
Başlangıç hareketi, SD kodları, 40, 43
BCL kodu, 620-622
BDT modeli, 10, 423, 425, 429
BDT sistemi, 10, 425, 424, 425, 428
BDT/BDÜ sistemleri, 11, 423, 428, 620
CADAM, 426, 543, 459
CATIA, 426, 429 CDS, 427
BDT/BDÜ sistemleri ile SD programlama,
422, 460
kesici hareketi, 492, 435
tartışmalar, 435, 438
örnek, 457
ilkeler 428, 435
işlem sırası (yöntemler), 440, 442, 444,
446, 456
Bekletme APT deyimi (DELAY deyimine
bakınız)
işleme çevriminde, 536, 537, 542
SD kodu, 38
Bilgisayar Bütünlüklü Üretim (BDÜ), 14, 620
Bilgisayar destekli SD programlama, 9, 11, 95
Bilgisayar destekli üretim (BDÜ), 3, 11, 15
423, 424
Bilgisayar grafikleri, 423
Bilgisayarlı Sayısal Denetim (BSD), 9, 17, 24
457, 559, 620, 621
Binary Coded Decimal (BCD) code, 20
Birim, ölçme, 294
Birim vektör, 161
BISECT değiştiricisi, 135
BLOCK DATA alt programı (DAPP soru
listesine bakınız)
değişkenler, 478, 489
BLOCK DATA altprogramı (DAPP soru
ünitesine bakınız)
Blok, SD, 19
Blok formatı, SD, 21, 26, 475, 485, 495
blok adresi, 21, 23
sabit sıralı, 21
tab sıralı, 23
kelime adresi, 23
Blok sonu kodu, 19, 25, 26
Boş karakter, APT, 103
Bölü işareti, 108
BRKCHP ikincil kelimesi, 542, 543, 467
BSD (Bilgisayarlı Sayısal Denetime bakınız)
Bunker-Ramo 3100 SD denetimcisi, 559
CALL deyimi, 324, 332, 333
CAM (Bilgisayar destekli üretime bakınız)
CANON ikincil kelime, 187, 195
deyim, 335, 341
CATIA sistemi, 426, 429
CBRTF fonksiyonu, 207
CCLW değiştiricisi, 115, 233, 304, 371,
480, 517
CENTER ikincil kelimesi, 124, 141, 162, 163,
180, 187
Ceplene, 370, 261
tanımlama, 370, 261
sapma yüksekliği, 370
CIM (Bilgisayar bütünlüklü üretime bakınız)
CIRCLE tanımlaması, 140, 155, 204, 205, 629
CLDATA, 10, 13, 96, 280, 528, 353, 398,
513, 412, 469, 552, 555, 577, 580,
619, 620, 621
içindekiler, 464, 469
format, 464, 465
çıkı, 352, 353, 361, 373, 375, 408, 411
çıkı deyimi (CLPRNT deyimine bakınız)
fiziksel gösterim, 465
kayıt tipleri, 464, 469, 515
standart (ISO standardı 3592-1978'e bakınız)
CLEARP deyimi, 520, 537
CLFILE, 398, 464, 469, 474, 619
kayıt tipleri (tablo), 466, 467, 468, 469
470, 471
CLKARG değişkeni (Değişkenler,
DAPP-tabanlı son işlemciye bakınız)
CLPRNT deyimi, 280, 311, 386
CLRSRF deyimi, 521, 537

- CLSEPT dizisi (Diziler, DAPP-tabanlı son
işlemciye bakınız)
CLT deyimi, 366, 466
CLW değiştiricisi, 115, 233, 304, 479, 480,
517, 526
COMPACT II, 10
CONST değiştiricisi, 230
Consumption, NC machine, 11
CONTIN deyimi, 322
COOLNT deyimi, 306, 492, 479, 521
COPY deyimi, 349, 356, 365, 389, 412, 526
COSF fonksiyonu, 207
COTANF fonksiyonu, 207
CROSS değiştiricisi, 161
CSINK ikincil kelime, 542
CUTOCOM deyimi, 481, 521, 593
CUTTER deyimleri, 513, 302, 468
OPTION, 300, 302
CYCLE deyimi, 480, 481, 487, 521, 535, 543
CYLNDR tanımlaması, 178, 181, 629
Çarpma (Sakınma, çarpımaya bakınız)
Çeviri Bölümü, IBM APT-AC SD İşlemcisi
(IBM APT-AC SD işlemcisine bakınız)
Çevresel İşleme Hareketi, APT, 225, 253, 395
365, 376
kontrol edici yüzeyler, 254, 259, 265, 366
368, 370
tanımlama, 259, 264, 376
yönler, 261, 262, 268
bitirme (ince işleme), 273, 274, 251, 370
işleme, 278, 280
programlama, 275, 278
kaba işleme, 273, 274
başlatma (başlatma hareketine, APT'ye
bakınız)
deyimler, 258, 264
bitirme, 264, 231
Çeşitli fonksiyon kodları (M koduna bakınız)
Çift bar işareti, 110
Çift dolar işareti, 104
Çift tırnak işareti, 110
Çift yıldız işareti, 108
Çizelgelenmiş silindir, 178, 630, 632
eğrilik, 630
tanımlama (TABCYL tanımlamasına bakınız)
örnekler, 393, 630, 632
çıkı, 187, 630, 632
Çıkı deyimi, APT, 203, 280, 311, 386
Çok denetim yüzeyleri, 251, 253
deyim, 366
TRANTO deyimi, 366, 253
Çok eksenli işleme, 449
D kodu, 25, 26, 40, 42, 60, 64, 66
Daire (CIRCLE tanımlamasına bakınız)
Dairesel enterpolasyon hareketi, 37, 38
306, 307, 567, 572, 601
Dairesel silindir (CYLNDR tanımlamasına
bakınız)
Dallanma deyimi, APT, 316, 319, 321
Dağıtılmış sayısal denetim (DSD), 9, 457
DAPP soru listesi, 488, 490, 500
cevaplar (örnekler), 500, 615
BLOCK DATA altyardam, 478, 500, 506
507, 508, 511, 475, 610
denetimci tanımlaması, 495, 498
takım tezgahı tanımlaması, 490, 495
çıkı formatı, 495, 499, 500
DAPP (Son işlemci için tasarım desteği), 474
DATBUF (Diziler, DAPP-tabanlı son işlemciye
bakınız)
DECR değiştiricisi, 236, 317
DEEP ikincil kelimesi, 542, 543, 572
DELAY deyimi, 306, 480, 522, 574, 601
Delikli şerit (Kağıt şeride bakınız)
DELTA değiştiricisi, 122, 232
Denetim Dallanma Deyimleri, APT, 316,
319, 321
Denetim (kontrol yüzeyi), 254, 256, 259
366, 370
tanımlama, 356, 366, 369
değiştirici, 259, 366, 369
çok, çoklu, 366, 369
Değişkenler, APT makro, 332 indisli, 316, 318
Değişkenler, DAPP-tabanlı son işlemci
CLKARG, 478, 488, 566
ilk değer, 490

- ISTART, 511, 512
 JLINE, 478, 489
 LENGTH, 511, 512
 NI, 511, 512
 OUTCTL, 478, 489, 550, 574, 607
 Değiştirici, 102, 115, 202
 denetim yüzeyi, 262, 263
 geometrik öğeler, 115, 119
 ölçüsel, 115, 117
 yönel, 115, 117
 bağunlu, 115, 117
 PATTERN deyimini, 240, 246
 kesici konumu, 259, 260
 Devir sayısı, SD'li tezgah, 12, 304
 Deyim boyutu, APT, 112
 DISTF fonksiyonu, 207, 228
 Dizi noktalar, 228
 Diziler, DAPP tabanlı son işlemci, 478, 488
 CLSETP, 515
 DATBUF, 478, 485, 511, 512, 513, 518,
 525, 532, 534, 535, 537, 543, 554, 560,
 562, 567, 568, 570, 571, 572, 574, 581,
 593, 601, 604, 605, 608
 ISTART, 511, 551
 LENGTH, 511, 551
 LIMITS, 478, 487
 LOGCAL, 478, 481, 485, 512, 521, 543
 OUTGRD, 478, 485, 486, 554, 601, 605
 TIME, 478, 488, 522, 524, 526
 Dış tolerans, 293
 DLWCIR yordamı, 482, 516, 533, 535, 567
 DLWCYC yordamı, 513, 516, 535, 543
 DLWDBF yordamı, 511, 594
 DLWDER yordamı, 554
 DLWLIN yordamı, 516, 531, 533
 DLWORG yordamı, 516, 528, 531, 532, 534,
 DLWPP1 yordamı, 516, 517
 DLWPP2 yordamı, 516, 517
 DLWROT yordamı, 516, 517, 526, 531, 532,
 534
 DLWSEG yordamı, 559, 562, 565
 DLWSRR yordamı, 559, 560, 562, 565
 DLWUP1 yordamı, 516, 517, 611
 DLWUP2 yordamı, 516, 517, 611
 DNTCUT deyimini, 366, 466
 DO deyimini, 322, 328
 Dolar işareti, 104
 Doğrulama, SD programı, 13
 Doğrultman, 474
 Doğrusal enetropolasyon hareketi, 36, 37, 531,
 533, 556, 567, 601
 DOTF fonksiyonu, 207, 228
 Döner tabla, 517
 Döngü programlama, 322, 332, 337
 Do döngüsü, 322, 328
 hatalar, 333
 LOOPST-LOOPND döngüsü, 328, 333
 Dönme yüzeyleri, 173
 Dönüşüm, 339, 365
 kesici yolu, 352, 356, 356-365, 394
 geometrik öğe, 349-352, 359-365
 Dönüşüm matrisi, 339-349
 DRILL ikincil kelimesi, 536
 Düzeltme bölümü, IBM APT-AC SD işlemcisi
 (IBM APT-AC SD işlemcisine bakınız)
 Düzenleme bölümü, son işlemci (işleme
 yordamları, son işlemciye bakınız)
 DWL değiştiricisi, 536, 537, 542
 E kodu, 26, 56, 59, 66
 EBCDIC, 544
 EIA kodu, 21, 582
 EIA standartları
 RS 258-C, 21, 459
 RS 271-B, 20, 21, 93
 RS 297-B, 20, 93
 RS 305-D, 463, 620, 622
 RS 400-B, 21, 93
 RS 545, 620, 621, 622
 Eksenler, SD'li tezgah, 19, 20
 Eksi işareti, 108
 Elektronik yüzey algılama, 12
 Elips, 161
 Elipsoid, 174
 Eliptik koni, 176
 Eliptik parabol, 174
 Eliptik silindir, 176

- Elle SD programlama, 17-81
 örnekler, 70-81
 ilkeler, 69-70
 ELLIPS tanımlaması, 161
 END deyimini, 308, 386, 480, 518, 574, 601
 Engeller, kesici yolu, 246
 EQ (bağlantı operatör), 231, 319
 Equal sign, 108
 Esnek üretim sistemleri (EÜS), 12, 620
 Etiket, APT deyimini, 110, 112
 EXAPT, 10
 EXEC (iş icra programı) son işlemci oluşturma,
 581
 BLOCK DATA altprogramını oluşturma, 506
 ADT programını işleme, 400
 EXPF fonksiyonu, 207
 F kodu, 25, 26, 37, 582, 601
 FANUC BSD denetincileri
 model 6MB, 17, 19, 25, 28, 48, 463, 554,
 567, 569, 572
 model 6T, 17, 19, 26, 29, 48, 54, 463, 551,
 554, 567
 Farklar (Kesici farkları, tanımlanmasına bakınız)
 FEDRAT deyimini, 227, 302, 303, 385, 479, 522
 FINI deyimini, 311, 324, 328, 333, 385, 466, 471
 FLOOD ikincil kelimesi, 306, 385, 479
 FORTRAN dili, 96, 400, 477, 580, 581, 610
 FROM deyimini, 227, 267, 270, 275, 469
 FUNOFY değiştiricisi, 165
 G kodları, 25-29, 582, 601, 603, 606
 G00, 28, 29, 35, 36, 43, 227
 G01, 28, 29, 36, 37, 43, 533
 G02 ve G03, 26, 29, 37, 38, 533
 G04, 28, 29, 38, 574
 G10, 28, 29, 42, 46
 G17, G18 ve G19, 28, 37-38
 G20 ve G21, 28, 29
 G28, 28, 29, 33
 G32, 29
 G40, G41 ve G42, 28, 29, 40-48
 G43, G44 ve G49, 88
 G50, 29, 32, 35, 463
 G70, 29, 60, 61, 78, 81
 G71, 60-62, 64, 78
 G72, 29, 60, 62, 64,
 G73, 29, 60, 64-66, 78
 G76, 29, 60, 66-67
 G80, 28, 48
 G81, 28, 48-53, 573
 G83, 28, 48, 53, 76
 G85, 28, 48, 53, 573
 G90, 28, 33, 55
 G91, 28, 35, 70
 G92, 28, 32-35, 56, 463
 G94, 28, 56
 G95, 28
 G98, 28, 29, 49
 G99, 28, 29, 49
 GCONIC tanımlaması, 161, 163-165, 629
 GE (bağlantılı operatör), 209, 319
 Genel işleme bölümü, son işlemci (işleme
 yordamları, son işlemciye bakınız)
 Genel konik (GCONIC tanımlanmasına
 bakınız)
 Geometrik modelleme, 425
 sınır tanımlaması, 426
 Yapısal Katı Geometri (YKG), 380
 katı, 425
 yüzey, 425
 tel çerçeve, 425
 Geometrik tanımlama, APT, 114-200, 349, 386
 bulanıklık, 200-205
 uygulama, 210-216
 GO deyimini, 265-268, 453, 469
 GOBACK deyimini, 259, 261
 GOCLER deyimini, 519, 537, 539
 GODLTA deyimini, 226, 469, 531
 GODOWN deyimini, 259, 261
 GOFWD deyimini, 259, 261
 GOHOME deyimini, 414, 518, 537
 GOLFT deyimini, 259, 261
 GORGT deyimini, 259, 261
 GOTO değiştiricisi, 249
 GOTO deyimini, 226, 228, 246, 271, 275, 453,
 469, 531, 534
 GOUP deyimini, 259, 261

- GT (bağlantılı operatör), 209, 319
 Hafıza kapasitesi, BSD denetimcisi, 9, 457
 Hareket tanımlaması APT (Çevresel işleme hareketi, APT, noktadan noktaya hareket, APT, başlangıç hareketi, APT) BDT/BDÜ sistemleri, 429, 435, 452
 SD kodları dairesel, 37, 38
 doğrusal enterpolasyon, 36, 37
 işleme çevrimi, 48, 67
 noktadan-noktaya, 35, 36
 Hareket yönleri, 261
 Hassasiyet BSD takım tezgahı, 12
 Hata ayıklama, APT programı, 383, 404, 408
 Hata numarası, APT, 404, 408
 Hazırlık fonksiyon kodları (G kodlarına bakınız)
 HIGH ikincil kelimesi, 528
 Hiperbol, 161, 163
 Hiperbolid, 174
 Hiperbolik parabol, 174
 Hızlanma/Yavaşlama zaman sabiti, 559
 Homojen dönüşüm, 341
 HYPERB tanımlaması, 163
 I kodu, 25, 26, 27, 37, 38, 43, 55, 64, 66, 568, 570, 571, 582, 603
 IBM ADT-AC SD işlemcisi, Release 4, 400
 IBM APT-AC SD işlemcisi, 12, 96, 400, 462, 47
 hesaplama bölümü, 360, 398
 denetim bölümü, 398, 399
 düzeltme bölümü, 400
 son işlemci bölümü, 398, 399
 çeviri bölümü, 398
 IF deyimini, 319, 324, 333
 IFAPT, 10
 IN değiştiricisi, 115, 117, 150
 INCHES ikincil kelimesi, 294, 470
 Inclusive subscripts, 317
 INCR değiştiricisi, 233, 317, 517
 INDEX deyimini, 356, 359, 363
 INDIRP deyimini, 268, 275, 369, 405
 INDIRV deyimini, 268, 275, 369, 405
 INSERT deyimini, 311, 522, 611
 INTERC değiştiricisi, 131
 INTGF fonksiyonu, 207, 324
 INTOF değiştiricisi, 126, 369
 INTOL deyimini, 293
 INVERS değiştiricisi, 244, 346
 INVX değiştiricisi, 240
 INVY değiştiricisi, 240
 IPM ikincil kelimesi, 303, 536, 542
 IPR ikincil kelimesi, 303, 536, 542
 ISO kodu, 21, 582
 ISO standartları ISO 83, 20, 93
 ISO 3592-1978, 464, 466, 622
 ISO 4343-1978, 464, 622
 ISO 6983/1-1982, 14, 463, 622
 ISTART dizisi (Diziler, DAPP tabanlı son işlemciye bakınız)
 İç tolerans, 293
 İşsel Sıra Numarası (ISN), 278, 466
 İki nokta üstüste işareti, 106
 İlerleme hızı, SD'li tezgah, 12, 302, 303
 İlerleme hızı tanımlaması APT, 302, 303
 SD kodları (F koduna bakınız)
 İletişim kanalları,
 sonişlemci, 477-489
 diziler (Diziler, DAPP-tabanlı son işlemciye bakınız)
 İlk değer atama bölümü takım tezgahı modülü, 551-552
 son işlemci, 471, 510, 511, 512
 İndislenirilmiş değişkenler, 316, 317, 318
 İzin verilen miktar, bitirme (ince), 228-230
 İş Denetim Dili (İDD), 384, 581, 633, 634
 İş icra programı (EXEC'e bakınız 400, 581
 İşleme çevrimleri
 APT devrimleri (CYCLE deyimine bakınız)
 SD kodları (Torna tezgahı) 54-67
 alın tornalama, 56, 62
 ince işleme, 61
 dizi tekrarlama, 64-66
 vida açma, 56-59, 66-67
 tornalama, 55, 60-62
 SD kodları (Freze tezgahı) 48-54
 delik büyültme, 48, 53
 havşa açma, 48

- delme, 48, 49
 gagalama delme, 48, 53, 76
 özel delik büyültme, 48, 53
 kılavuz delme, 48
 İşleme eksenini, 7
 İşleme özellikleri, 67, 69, 102, 292, 554
 İşleme yordamları, son işlemci, 469, 474
 İşletim sistemleri
 OS/MVS, 384, 400, 633, 634
 VM/CMS, 384, 400, 580, 581
 J kodu, 25, 37-38, 43, 568, 570, 571
 JUMPTO deyimini, 319, 321, 324, 333
 conditional, 321
 unconditional, 319
 K kodu, 25, 26, 37-38, 43, 56, 64, 66, 568, 570, 571, 582, 603
 kanonik biçimler, 115, 200, 629
 tartışma, 200-205
 değiştiricisi, 115-119
 işlem sırası, 212
 deyim formatı, 112
 dönüş yüzeyleri, 173
 Kanonik formlar, 115, 200, 629
 parametre çıkarımı, 110, 211
 Kağıt şerit, 9, 21, 457, 478, 544
 Karekök (SQRTF fonksiyonuna bakınız)
 Katı model, 425
 Kayıt, takım farkı, 42
 Kelime adresi, 24
 Kelimeler, APT, 98, 99-102, 626-628
 bilgisayar işlemi, 100
 kesici hareketi, 100
 sabit alan, 102
 geometrik öğeler, 100-101
 esas, 102, 114
 matematiksel işlem, 100
 ikincil, 102, 114
 son işlemci, 100-101
 işlem denetimi, 309-311
 Kesici değişimi, otomatik, 7, 304
 Kesici eksen çizgisi bilgisi (CLDATA'ya bakınız)
 Kesici farkı, tanımlama, 46-47
 Kesici konum değiştiricisi, 259-260
 kontrol yüzeyi, 259, 263
 tahrik yüzeyi, 259-260
 parça yüzeyi, 259-260
 Kesici tanımlama, 295-297
 typhotetic disk, 301-302
 hareket yönü, doğrultusu, 261
 parametreler, 295-297, 468
 konum tanımlaması, 294
 üst yüzey etkisi, 297-302
 Kesici uç yarıçapı, 46, 47
 Kesici yarıçapı telafisi (Telafiye bakınız)
 Kesici yerleşim bilgisi (CLDATA'ya bakınız)
 Kesme sıvısı (Soğutucuya bakınız)
 Kişisel bilgisayar, 457
 kod, 26, 60, 582
 Koni, 187
 Koni tanımlaması, 187
 Koordinat sistemi tanımlama, 19, 32-35,
 308-309, 384 APT'de, 308-309, 384, 523,
 528-531
 SD kodlarında, 32-35
 SD'li tezgahlar, 19-20, 308-309, 384
 parça, 277-278 tornalama işlemi, 19, 32-35
 304, 384
 Kuadrik yüzey, 173
 Kümelenmiş döngüler, 324
 Kümelenmiş geometrik tanımlama, 106-108
 Küp kök fonksiyonu (CBRTF fonksiyonuna bakınız)
 L kodu, 25, 26, 49, 53, 67, 582
 LARGE değiştiricisi, 115, 117
 LAST değiştiricisi, 355
 LE (bağlantılı operatör), 231, 319
 LEADER deyimini, 522
 LEFT değiştiricisi, 522
 LENGTH dizisi (Diziler, DAPP tabanlı son işlemciye bakınız)
 LENGTH ikincil kelimesi, 158
 LIMIT dizisi (Diziler, DAPP-tabanlı son işlemciye bakınız)
 LINE deyimini, 129-140, 629
 LINEAR ikincil kelimesi, 235

- LNTHF fonksiyonu, 207
LOADTL deyimi, 304-306, 395, 475, 479, 488, 522, 594, 595, 596
Loft (yüksek) konikler, 165
LOG10F fonksiyonu, 207, 228
logaritmik fonksiyon, 228
LOGCAL dizisi (Diziler, DAPP-tabanlı son işlemciye bakınız)
LOGF fonksiyonu, 228
LOOPND deyimi, 328, 333
LOOPST deyimi, 294, 299
LOW ikincil kelimesi, 474LT (bağlantılı operatörler), 209
M kodları, 25, 26, 29, 67-68, 574, 582, 594, 595, 599, 601, 600
MACHIN deyimi, 310, 386, 398, 443, 465, 481, 523, 583, 594
MACRO deyimi, 324, 325, 328, 332
Maliyet, son işlemci, 619
Manyetik kaset teypi, 457
Matematiksel fonksiyonlar, APT, 227-232
Matematiksel hesaplama, APT, 227-232, 337-338
Matris, 339-349
affine transformation, 341
genel dönüşüm, 340
dönüş, 340
çeviri, 339
Matris bölümü, 346
MATRIX tanımlaması, 341-349, 629
MAXIF fonksiyon, 228
Mazatrol CAM M-2 BSD denetimcisi, 11
MCHTOL deyimi, 480, 523
Mekatronik, 12
MILL ikincil kelimesi, 542
MINIAPT, 10
MINIF fonksiyonu, 228
MINUS ikincil kelimesi, 161
MIRROR ikincil kelimesi, 344, 345
MIST ikincil kelimesi, 306, 385, 479
MM ikincil kelimesi, 294, 470
MODF fonksiyonu, 228
MODIFY değiştiricisi, 241, 244, 356, 357
Modüler tasarımı, SD'li tezgah, 7
MTM işleme dairesel enterpolasyon hareketi, 567-572
doğrusal interpolasyon hareketi, 556-567
işleme çevirileri, 572-574
işleme özellikleri, 554, 574
hareket kodu, 554-574
parçalama, doğrusal hareket, 559-564
özel fonksiyonlar, 574
MTM (Takım Tezgahı Modülüne bakınız)
MULTAX deyimi, 480
N kodu, 25, 26, 582
NE (bağlantılı operatör), 209
NEGX değiştiricisi, 115
NEGY değiştiricisi, 115
NEGZ değiştiricisi, 115
NELAPT, 10
Nokta vektörü, 155
Noktadan-noktaya hareket, APT, 225, 226-253
GODLTA deyimi, 226
GOTO deyimi, 226
GOTO/(dizi) deyimi, 228, 246
Noktadan-noktaya hareket, SD kodları, 35-36
Noktalama, APT (APT dili, noktalamaya bakınız)
Noktalı virgül, 104
NOMORE ikincil kelimesi, 350, 351, 352, 355, 356, 359
NOPOST deyimi, 310
NOPS deyimi, 266
NORMAL ikincil kelimesi, 183, 184, 185
Normaller, çizelgelenmiş silindir, 182, 186
Notasyon, APT deyim formatı, 112
NOX ikincil kelimesi, 232
NOY ikincil kelimesi, 232
NOZ ikincil kelimesi, 232
NUMPTS deyimi, 278
NUMPTS ikincil kelimesi, 233
O kodu, 25, 26, 67
OBTAIN deyimi, 234
OFF ikincil kelimesi, 300, 301, 304, 306, 307, 335, 386, 479, 480, 523, 542
OFFSET deyimi, 369

- OMIT değiştiricisi, 233, 242, 244
ON ikincil kelimesi, 122, 259, 262, 274, 304, 306, 307, 335, 341, 386, 465, 479, 480, 523, 537
OPSKIP deyimi, 523
OPSTOP deyimi, 480, 518, 601
OPTION ikincil kelimesi, 300-302, 310, 523
ORIGIN deyimi, 308-309, 474, 516, 523
Orijin, SD'li tezgah (Tezgah orijinine bakınız)
Otomatik oluşum, SD programı, 13, 435-438
OUT değiştiricisi, 115, 117, 150
OUTCTL değişkeni (Değişkenler, DAPP-tabanlı son işlemciye bakınız)
OUTGRD dizisi (Diziler DAPP-tabanlı son işlemciye bakınız)
OUTTOL deyimi, 293
Overshoot, 556-558
Örnekler APT programı, 387-397, 611
SD programı, 18, 53, 69-81
P kodu, 25, 26, 38, 42, 60, 64, 67-68, 582
Palet sistemi, 12
Parametrik eğri, 428
Parantez, 106
Parça programı, APT (APT dili, programına bakınız)
Parça yüzeyi, 254, 256, 265, 370-371
tanımlama, 256, 370-371
başlangıç hareketi, 265, 369
deyim, 265
Parity, 21
Park konumu, SD'li tezgah, 493, 495
PARLEL değiştiricisi, 131
PARTNO deyimi, 309, 385, 456, 465, 524, 544
PASCAL dili, 96
PAST değiştiricisi, 259-260, 262, 265
PATTERN tanımlaması, 228-239, 317, 534
Periyod işareti, 110
PERPTO değiştiricisi, 133
PLABEL deyimi, 524, 544
PLANE tanımlaması, 165, 168-172, 629
PLUS ikincil kelimesi, 161
PNTVCT tanımlaması, 155
POCKET deyimi, 370
POINT tanımlaması, 120-129, 202, 629
POLCON tanımlaması, 195-197
Polikonik yüzey, 192-195, 420
POSX değiştiricisi, 115
POSY değiştiricisi, 115
POSZ değiştiricisi, 115
PPRINT deyimi, 311, 524
PREFUN deyimi, 524, 574, 601
PRINT deyimi, 203, 386
PROFIL, 398, 464
Program yapısı, APT, 385-387
PSIS deyimi, 271
PSTAN değiştiricisi, 259, 263
PTNORM ikincil kelimesi, 182, 183
PTSLOP ikincil kelimesi, 182, 183
Q kodu, 25, 26, 53, 60, 64, 582
QADRIC tanımlaması, 173, 629
R kodu, 25, 26, 37, 42, 48, 49, 53
RADIUS ikincil kelimesi, 121, 141
RAPID deyimi, 227, 303, 385, 518, 534, 539
Referans noktası, SD'li tezgah, 32
REFSYS deyimi, 349-352, 359-361, 398
REMARK deyimi, 311
RESERV deyimi, 317, 328, 386
RESET deyimi, 519
RETAIN ikincil kelimesi, 233, 244
RETRCT deyimi, 519, 537
RIGHT değiştiricisi, 115, 117
RLDSUF tanımlaması, 189, 192
ROM (Sadece okunabilir hafıza), 7
ROTABL deyimi, 479, 480, 516, 517, 524, 526
ROTREF ikincil kelimesi, 356, 526
RS232 arabirim kablosu, 459
RTHETA ikincil kelimesi, 121
S kodu, 25, 26, 68, 582, 599
Sakınma, çarpma, 246
SAME ikincil kelimesi, 356
Sapına dairesel yol, 9, 457
Sapına yüksekliği, 370, 429
Sayısal denetim (SD) uygulama, 11
varolan durum, 11-13
tanımlama, 3
tarihçe, 3-11

- tasarım ve üretime etkisi, 13-15
tezgahlar, 5-9, 11-12
eğilim, 13
SCALE ikincil kelimesi, 344, 346, 356, 357
SD işlemcisi (IBM APT-AC SD işlemcisi)
SD kodları, 17, 20-21, 28-31, 582
format (Blok formatı, SD'ye bakınız)
SD Program oluşumu otomatik (otomatik oluşum, SD programına bakınız) ilkeler, 423, 428-435
işlem sırası, 440, 442-444, 446-456
SD (Sayısal kontrole bakınız)
SELCTL deyimi, 306, 479, 488, 524
Semboller, APT, 98, 102, 119
SEQNO deyimi, 480, 524
SETOOL ikincil kelimesi, 304-305, 475, 522
SIGNF fonksiyonu, 219
Silindirik yüzeyler, 173, 178
SINF fonksiyonu, 207
Sıfır, 187
Sınırlandırılmış yüzey, 188, 189
Sınırlandırma, 188
Skalar ürün, vektörler, 207
Skalarlar, 102
SLOPE ikincil kelimesi, 131, 183, 186
SMALL değiştiricisi, 115, 117
Son işlemci, 10, 13, 96, 462, 618, 619-622
çağırma deyimi (MACHINE deyimine bakınız)
iletişim (iletişim kanallarına, son işlemciye bakınız)
tasarım örneği, 581-617
tasarım yöntemleri, 580-581
programlamaya etkisi, 384-385
fonksiyonları, 474-475
oluşturucu, 13, 622, 623
uyarlama, 462, 580-581, 610-611
gereklilik, 463-464
çıktı, 611, 616
işlem akışı, 577
işleme yordamları (işleme yordamları, son işlemciye bakınız)
yapı, 469, 471-474, 577
deneme, 611
eğilim, 13, 619-622
Son işlemci bölümü, IBM APT-AC, SD işlemcisi, 398, 399
Son işlemci kelimeleri (APT diline bakınız)
Son işlemci oluşturucusu, 13, 622, 623
Soğutucu, 306
SPALIN tanımlaması, 156
SPHERE tanımlaması, 187-188
SPINDL deyimi, 304, 479, 525
SPLINE ikincil kelimesi, 112
SQRTF fonksiyonu, 207
Standart kesici uç vektörü, 46
STEP ikincil kelimesi, 187
Stick-slip etkisi, 6
STOP deyimi, 480, 518, 574, 601
T kodu, 26, 47, 68, 88, 90, 582, 583, 596
TABCYL tanımlaması, 178, 182-187, 212, 630-632
Tabii logaritma (LOGF fonksiyonuna bakınız)
Tahrik yüzeyi, 254, 256, 259, 265, 468
çevresel işleme, 256, 259
tanımlama, 254, 256
başlangıç hareketi, 265
Takım Tezgahı Modülü (MTM), 474, 477, 544, 546-579, 576, 583
denetim yönlendirme bölümü, 547, 550, 607-609
hazırlık bölümü, 547, 550, 603-606
işleme bölümü, 547, 550, 554-574, 594-602
yapı bölümü, 547-550
değişken tanımlama bölümü, 547, 548, 585-586, 593
TANF fonksiyonu, 207
TANTO değiştiricisi, 136, 259, 263
TAP ikincil kelimesi, 536
TAPKUL ikincil kelimesi, 306, 385, 479
Tasarım, SD'li takım tezgahı, 293
Taşınabilirlik, SD programı, 619, 620, 621
Tekrar edilebilirlik, SD'li tezgah, 12
Tekrarlı işleme çevrimi (işleme çevrimlerine bakınız)
Tel çerçeve model, 425

- Telafi kesici uç yarıçapı, 46, 47
kesici yarıçapı, 40-48
TERMAC deyimi, 324, 332, 333
Tezgah Denetim Ünitesi (TDÜ), 7
Tezgah orijini, 308
tezgah-bağımlı (Takım Tezgahı Modülüne bakınız)
tezgah-bağımsız, 520-546
genel işleme bölümü, 471, 478, 515-545
düzenleme bölümü, 471, 512-513, 609
ilk değer atama bölümü, 471, 511-512
çıkırtı bölümü, 473, 543-544
okuyucu bölümü, 471, 514-515
THETAR ikincil kelimesi, 121
THICK deyimi, 273, 335, 366
THRU ikincil kelimesi, 144, 246, 317
TIME dizisi (Diziler DAPP-tabanlı son işlemciye bakınız)
TIMES ikincil kelimesi, 124, 159
TITLE deyimi, 102
TLLFT değiştiricisi, 259-260
TLOFPS değiştiricisi, 259-260
TLON değiştiricisi, 259-260
TLONPS değiştiricisi, 259-260
TLRGT değiştiricisi, 259-260
TO değiştiricisi, 259-260, 262
Tolerans deyimleri, APT, 278, 292-294
INTOL, 293
OUTTOL, 293
TOLER, 293
Torna tezgahı,
CADILLAC NC-110, 76, 500, 581
BLOCK DATA alt programı, 507-508
özellikler, 582-583
DAPP soru listesi cevapları, 502-506
Takım Tezgahı Modülü, 583-609
son işlemci, 581-610
LCONIC tanımlaması, 161, 165-167, 212, 318, 387-390, 629
TORUS tanımlaması, 187
TRACUT deyimi, 349, 352-356, 359-361, 398, 412, 526
TRANPT ikincil kelimesi, 240
Transfer, SD programı, 457-459
TRANSL ikincil kelimesi, 240, 244, 343, 347, 356, 357
TRANTO deyimi, 366
TREND SD işleme, 619-622
SD programlama, 13, 435-438
SD teknolojisi, 11-15
TRFORM ikincil kelimesi, 182, 183
Trigonometrik fonksiyonlar, 207
TYPEF fonksiyonu, 207
U kodu (Torna), 26, 35, 38, 55, 60, 64, 582, 603, 606
UNIT ikincil kelimesi, 135, 161
UNIT'S deyimi, 294, 470
Uzay doğrusu, 156
Üretim, SD'li tezgah, 11
ÜSSEL fonksiyon (EXPF fonksiyonuna bakınız)
VECTOR tanımlaması, 155-161, 629
Vektör, 155
Virgül işareti, 110
W kodu (torna), 25, 35, 55, 60, 64, 582, 603
Wankel motoru, ...in profili, 212, 387
X kodu, 25, 26, 38, 49, 55, 582, 606
XAXIS ikincil kelimesi, 130
XCOORD ikincil kelimesi, 232
XLARGE değiştiricisi, 115-117
XSMALL değiştiricisi, 1115-1117
XYPLAN ikincil kelimesi, 121, 344, 520, 521
XYROT ikincil kelimesi, 343, 346, 356, 357
XYZ ikincil kelimesi, 232
Y kodu, 25, 49
Yapısal konfigürasyon, SD'li tezgah, 6-7
YAXIS ikincil kelimesi, 130
YCOORD ikincil kelimesi, 232
Yıldız, 108
YLARGE değiştiricisi, 115-117
Yorumlar, APT programı, 311
YSMALL değiştiricisi, 115-117
Yüzey model, 425
YZPLAN ikincil kelimesi, 121, 344, 520, 521
YZROT ikincil kelimesi, 343, 346, 356, 357
Z kodu, 25, 26, 48, 49, 53, 55, 56, 582
ZCOORD ikincil kelimesi, 232

ZLARGE deęiřtirici, 115-117
 ZSMALL deęiřtirici, 115-117
 ZSURF deyim, 120, 199-200
 ZXPLAN ikincil kelimesi, 121, 344, 520, 521
 ZXROT ikincil kelimesi, 343, 346, 356, 357

TERİMLER SÖZLÜĐÜ

Altprogram : Bilgisayar hafızasında okunan ve saklanan ve bir çağırma deyimini ile çağrılana kadar icra edilmeyen, iş parçasını işleme programının bir parçası.

Akış şeması (flowchart) : Bilgisayar programlamasında, programa uygun olarak yürüteceęi işlemlerin nitelik ve sırasını şematik olarak gösteren diyagram.

APT (Automatically Programmed Tool) : Sayısal denetim programını basitleştirmek için derleyici yazılım (software). APT, bu amaç için çok yaygın şekilde işletici (processor) olarak kullanılır.

Blok : Bir satır meydana getiren deyim veya komut.

Bilgisayar grafikleri : Giriş birimini geometri ve dięer vasıfları belirlemede ve çıkış birimini resim göstermede kullanarak bir bilgisayarı resim çizmede kullanılmaktadır. Bu, mühendislere geometri yoluyla bilgisayarla haberleşme imkanı verir.

Bilgisayar Destekli Tasarım (CAD) - (Computer - aided design) : Bilgisayarı verimlilik aracı olarak kullanarak tasarımın kendisinin yaratılması ve optimizasyonudur. CAD'in elemanları; bilgisayar grafiklerini, kullanıcı arabirimini, analizleri ve geometrik modellemeyi içerir.

CNC (Computer numerical control) : Bilgisayarlı Sayısal Denetimin baş harflerinden meydana gelmiştir. Buradaki bilgisayar (genellikle mikro) denetleyicinin bir parçasıdır ve tek bir CNC tezgahına tahsis edilmiştir.

Deyim (statement) : Kaynak programı oluşturan ve bilgisayarın bilgileri nasıl işleyeceğini tanımlayan komutlar.

Denetleyici (MCU) : Bir takım tezgahının hareketlerini kontrol eden aparat. Denetleyici tahsis edilmiş bir bilgisayarı içerebilir ve içermeyebilir.

Deęiřtirici (modifier) : Bilgi işleminde: bir komuta gerçekten uymayı sağlamak için, komutu önceden belirlenen bir biçimde deęiřtirmek için kullanılan bir nicelik; örneğin bir işlenenin adresini deęiřtirmek için kullanılan çevrim indisi gibi bir nicelik.

Dönen parça : İş parçasının döndürülmesiyle yapılabilen parça. Genellikle bir diřlide olduęu gibi bir eksene göre simetriktrir.

Eğri : Uzayda doğru veya eğri bir hat.

Geometri : Çizgilerin, açıların, yüzeylerin ve katı cisimlerin ölçümüyle uğraşan matematiğin bir kolu.

Geometrik modelleme : Geometrik cisimleri tanımlamada kullanılan geometri tekniği.

İşlem kodu (operation code) : Bir iş istasyonunda veya bir makinada yapılan bir işlem serisinin gösterilmesi için kullanılan sayısal kod.

İşlem planı (operation plan) : Bir işlem kodu tarafından gösterilen işlemlerin ayrıntılı tanımı.

İmal edilebilirlik (manufacturability) : Bir ürünün imal edilebilir olup olmadığının değerlendirilmesi (saptanması).

İlerleme hızı (feedrate) : Parça profili boyunca hareket eden veya bir noktadan diğer bir noktaya hareket eden kesici takımın hızına denir.

Kaydedici (Register) : Bir bilgisayarın işlem yapabildiği baytların (8 bitlik gruplar) sayısına dayanan ikili sayı grubu. Kaydedici bitlerin donanım yoluyla gruplandırılmış halidir.

Kenar : Bir yüzeyin bir yanını sınırlayan eğri

Kontrol edilebilir faktörler : Üretim, işlemi esnasında, boyutlar, toleranslar ve malzeme tipleri gibi kontrol edilebilen elemanlar.

Makro : Tek bir emirle başarılı şekilde gösterilebilen bilgisayar talimat grubu; örneğin; birçok noktanın delinmesi.

NC (Numerical Control) : Otomatik makineleri, özellikle takım tezgahlarını denetlemek için sayısal bilgisayarların kullanıldığı sistemleri tanımlayan terimin baş harfleri.

Nitelik kodu (attribute code) : Her parça niteliği, bir kod'da sabit bir konuma atanmıştır. Bir koddaki her karakterin anlamı, başka bir karakterin değerinden bağımsızdır.

Noktadan noktaya : Ayrı ayrı noktalarda yapılan tezgah işlemleri; delme ve kılavuz işlemleri tipik örnekleridir.

Tel çerçeve : Yüzeyleri içermeksizin parçanın sadece kenarlarını ve uç noktalarını belirleyerek bir cismin tanımlanması.

Tel çerçeve model : Bir parça ya da nesnenin üç boyutlu uzayda tanımlanması için bilgisayar grafiklerinde kullanılan en basit metod.

Transformasyonlar : Cisimlerin, matrik cebri kullanılarak matematiksel olarak çevrilmesi, döndürülmesi ve ölçeklendirilmesidir. Transformasyonlar, cisimleri bir ekranda hareket ettirmek için kullanılır.

Sayısal kodlama (numeric coding) : Bilgileri, sayısal biçim veya kodlara indirgeyerek makinenin kabul edilebileceği biçimde hazırlamak için kullanılan kısaltma sistemi.

Sayısal veri (numeric data) : Sayılarla gösterilmiş veri.

Sınırlayıcı (delimiter) : Bir karakter veya veri dizisini sınırlayan ve bu nedenle de diziye ait olması mümkün olmayan bir karakter.

Son işlemci (postprocessor) : Özel bir takım tezgahı için jenerik CL verilerini çeviren bilgisayar programı.

Sürekli sinyal (continous signal) : Sürekli gerilim sinyali gibi bir işlemin belirgin kesintisi olmayan veri takımı.

Sürekli kesme : Ayrı ayrı noktalardan ziyade talaş kaldırma işleminin sürekli olmasıdır. Freze, tipik bir sürekli işlemdir.

Parça tanımı : Tasarım geometrisinin tezgah işleme karşılığı.

Performans değişkenleri : Bir parçanın işlemlerini tanımlayan parametreler. Performans değerleri; tasarımcı tarafından parçanın tatminkar olarak çalışıp çalışmadığını ölçmek için kullanılır.

Yüz : Katı bir cismin bir yanını tanımlamada kullanılan sonsuz yüzeyin sınırlı bir bölgesi.

Yüzey : Çoğunlukla x, y ve z eksenlerinde bir denklem takımıyla tanımlanan düzlemsel veya şekillendirilmiş kesiksiz (sürekli) bir tabaka (levha)

Vektör : Hem büyüklüğü ve hem de yönü bulunan, ucunda bir ok işareti taşıyan bir doğru ile gösterilen bir nicelik.

ÖĞRETMEN MARŞI

Alnımızda bilgilerden bir çelenk,
Nura doğru can atan Türk genciyiz.
Yeryüzünde yoktur, olmaz Türk'e denk;
Korku bilmez soyumuz.

Şanlı yurdum, her bucağın şanla dolsun;
Yurdum, seni yüceltmeye antlar olsun.

Candan açtık cehle karşı bir savaş,
Ey bu yolda ant içen genç arkadaş!
Öğren, öğret hakkı halka, gürle coş;
Durma durma koş.

Şanlı yurdum, her bucağın şanla dolsun;
Yurdum, seni yüceltmeye antlar olsun.

İsmail Hikmet ERTAYLAN

